CHAPTER

,2

1 Towards Unsupervised Multi-Resident Tracking in Ambient Assisted Living: Methods and Performance Metrics

Tinghui Wang[,a,*] **and Diane J. Cook**[*]

[*]*Washington State University, School of Electrical Engineering and Computer Science, Pullman, WA*

CHAPTER OUTLINE HEAD

ABSTRACT
Ageing is a global challenge facing our society in the next few decades. Ambient assisted living (AAL) is a promising technology that helps people stay active, socially connected and independent into older age. Even though ambient binary sensors, such as PIR motion sensors, offer a low cost, easy to deploy and less intrusive solution to constructing a smart environment, the limited ability of coping with multiple residents hinders the wide adoption of the AAL technology. In this work, we present three multi-resident tracking algorithms, NN-SG, GNN-SG and sMRT, to solve the data association problem between the ambient sensor events and residents in the smart environment. We also introduce new performance metrics to evaluate the success of alternative approaches to multi-tracking tracking in smart homes. We evaluate all the algorithms with a recent smart home dataset recorded in real-life settings. Among the three algorithms, NN-SG and GNN-SG require sensor location and floor plan of the environment to derive the sensor graph, while sMRT does not require such information and relies solely on the unannotated sensor data. As an initiative of the unsupervised resident tracking solution, sMRT prompts additional research opportunities in multi-resident tracking to improve the adoption of AAL technology in our daily life.

Keywords: Smart home, multi-resident tracking, probability hypothesis density filter, Gaussian mixture

1.1 INTRODUCTION

Given recent medical advances in our society, people today are living longer and generally healthier lives. According to the United Nations world population prospect, by 2050, one in six people in the world (16%) and more than one in four people in more developed regions (27%) will be over 65 years old [1]. The aging of the population represents the great achievements of our medical and technological advances. However, at the same time, it poses dramatic challenges to society. Ambient assisted living (AAL), which introduces information and communication technologies to assist with a person's daily living and working environment, is a promising solution to help people stay active, socially connected, and independent into older ages [2]. By monitoring the daily activities of the residents via sensor networks, AAL environments may acquire the intelligence to recognize the residents' activities, monitor their well-being, and provide assistance and intervention when needed.

In the past, most research in AAL and smart homes focused on mono-occupant settings, where the smart home or the environment is inhabitant by single individuals. Based on the data collected by the sensors deployed in the AAL environments and smart homes, data-driven or ontology-based methods have been proposed to recognize activities of daily living (ADLs), understand resident intent, forecast their future activities, monitor and assess physical and mental health status, and enable building automation to minimize energy consumption. However, the ability to handle the multi-resident scenarios hinders wide-spread real-life adoption of AAL technology [3].

There are two significant challenges in a multi-resident smart home and AAL environment: resident tracking and resident identification [4]. The objective of resident tracking is to associate the data collected by the sensors deployed in the environment with the corresponding residents in order to monitor and provide fine-grained location-based services to the elderly living in the environment. The resident identification then tries to distinguish residents from one another based on the data association identified by the resident tracking process. In this chapter, we focus on the resident tracking problem.

In past decades, many researchers have proposed different tracking algorithms and sensor technologies to cope with the multi-resident scenario. Sensors used for multi-resident tracking involves video-based camera system [5], smart floor [6], passive infrared (PIR) motion sensors [7, 8, 9, 10, 11], RFID-based and Wifi-based system [12], and other ultrasonic systems [13]. Among those sensor technologies, PIR motion sensors offer a low cost, easy to deploy, and unintrusive solution. As the data collected by PIR motion sensors can not identify the resident who activates the sensor, the data association problem in the multi-resident environment is a popular research topic that interest many researchers. Some resident tracking solutions proposed in the literature assume that the number of residents in the space is constant. However, in reality, the number of residents may change when neighbors, family members, friends or care givers come and visit the resident. Moreover, if there are pets in the household, the activity of the pets may trigger the ambient sensors, which results in a multi-occupant scenario even in a single-resident smart home. Other research addresses the resident tracking problem by taking advantage of the additional information, such as annotated labels, the physical models of the sensors, floor plan of the environment and the location of the sensors deployed in the environment. However, such information may be impractical or too costly to obtain in real-life deployment. Thus, developing a resident tracking solution that could solve the data association problem solely from the sensor data without expert annotation or additional information would advance the adoption of smart home technology in real life.

To study the multi-resident tracking problem, we present three algorithms in this chapter: nearest neighbor with sensor graph (NN-SG), global nearest neighbor with sensor graph (GNN-SG) and multi-resident tracking with sensor vectorization (sMRT). NN-SG and GNN-SG, both of which are extended from the GR/ED method proposed by Crandall and Cook [9], can handle a varying number of residents in the smart environment, while the corpus of information comes from extra knowledge of physical sensor locations and a floor plan of the AAL environment. In contrast, the sMRT algorithm, proposed in this work, is the first attempt to solve the data association problem of multi-resident tracking using the sensor data alone without any additional information. Instead of requiring a floor plan and sensor map of the environment, sMRT learns the spatio-temporal relationship between sensors from unlabeled sensor data, and applies a multi-target Gaussian mixture probability hypothesis density (GM-PHD) filter to estimate the resident state, solve the association between sensor event and resident and estimate the number of active residents in the environment simultaneously.

To validate the approach, we evaluate all three methods using data collected from an actual smart home with ground truth labels for resident data association. In addition to the multi-class classification metrics commonly used in prior research, we also evaluate their ability to estimate the number of active residents in the smart home. Finally, we propose a multi-resident tracking accuracy (MRTA) score to further diagnose algorithm tracking errors.

The chapter is organized as follows. Section 2 provides a summary of previous research work in multi-resident tracking with ambient motion sensors. Section 3 details the sensor data collected in a smart home environment, and introduces the dataset used for evaluation in this work. NN-SG, GNN-SG and sMRT algorithms are proposed in Section 4, with performance metrics described in Section 5 and results and discussion presented in Section 6. Section 7 presents the conclusion and Section 8 offers future research directions in multi-resident tracking.

1.2 CHALLENGES AND RELATED WORK

Smart home technology combines sensor technology and artificial intelligence to provide various services and applications in assisted living environments for the elderly, including health monitoring, cognitive assessment, location-based personal services, and home automation. While passive ambient sensors offer an unobtrusive technology for monitoring the daily routine of smart home residents, however, these sensors lack the ability to identify the resident when activity is detected. Consider the case when one resident is moving around in the bedroom cleaning, while another resident is cooking in the kitchen, the sensors in the bedroom and the sensors in the kitchen will activate at the same time. As a result, the sensor events generated by both residents will be merged into the sensor event stream of the smart home in chronological order. As the sensors are anonymous, a resident tracking algorithm has to be introduced to segregate those sensor events into multiple tracks, each corresponding to one resident in the smart home. For example, in the case above, one track will be composed of the sensor events reported by the sensors in the bedroom while another track consisted of the sensor events of the sensors deployed in the kitchen. Moreover, the joint activity performed by multiple residents in the smart home further complicates the problem, as the correspondence between the residents and the sensor events may not be a simple one-to-one association. When two residents walk together from the kitchen to the dining room, the sensor events triggered along the way are associated with both residents at the same time, triggering a one-to-many association.

Recently, researchers have been studying the multi-resident tracking problem. In these works, the multi-resident tracking problem is commonly formulated as a data association problem between sensor events and the residents in the smart home. However, depending on the assumption of information availability, the solution and performance vary dramatically. Some work assumes that the floor plan and sensor locations of the smart home site are readily available. Other works assume that the dataset contains ground truth labels of resident association with sensor events or activities, so that the activity and mobility models of each resident can be learned using data-driven methods. In this section, we provide a summary of each of these research directions.

Wilson and Atkeson [11] and Hsu, et al. [10] combine the multi-resident tracking with the activity recognition, and propose a solution that solves both tasks at the same time. In both works, the number of residents in the smart home is specified a priori and remains constant, and the resident activity labels are provided. Based on the annotated data, Wilson and Atkeson [11] constructs a hidden Markov model (HMM) in which the hidden states represent the combination of the resident activities and the resident locations, and the observable states are mapped to the sensors deployed in the smart home. Thus, the data association problem is equivalent to the inference problem of the constructed HMM and can be solved with a Rao-Backwellised particle filter. Hsu, et al. [10] construct three conditional random fields (CRF) to model the relationship between activities of interest, residents in the smart home and the sensors deployed in the smart home. The data association is solved using an iterative inference algorithm.

With the number of residents in the smart home as a constant, Crandall and Cook [14, 15] consider the data association between sensor events and smart home residents as a multi-class classification problem. Thus, based on data with ground truth labels, a naive Bayes classifier and a Markov model classifier are both trained to predict the associated resident with a series of sensor events as the input. Their work concludes that subtle differences exist and can be learned using supervised learning algorithms to identify the associated resident.

In real-life settings, the number of residents who are actively performing daily activities in the smart home is not a constant. A family friend, relative or caregiver may visit, leading to an increase of the number of residents in the smart home than previously assumed. On the other hand, one resident may taking a nap at part of the smart home and remain undetected for a period of time, and the number of active resident in the smart home decreases. In order to cope with the varying number of active residents, other research focuses on constructing a model of resident dynamics in the smart home. A sensor graph [7], also referred to as Bayes updating graph [9], or accessibility graph [16], is a standard graph model that captures the resident movement information in the smart home. In the graph, the nodes are mapped to the sensors deployed in the smart home. The sensors that are physically adjacent to each other in the smart home are connected, and a weight can be assigned to each edge representing the likelihood of the resident moving from one sensor location to the other. The sensor adjacency information can be collected from the floor plan and the location of the sensors in the smart home or by conducting a controlled experiment in the smart home. The weights, however, can be estimated using annotated data or maximizing the likelihood of a recorded sensor event stream [16]. With the sensor graph constructed, the multi-resident tracking problem can be solved with a rule-based tracker [9] or a multi-hypothesis tracker [7].

In addition to sensor adjacency, a detailed model of the field of view (FoV) of each sensor with respect to the floor plan of the smart home can provide valuable information to solve the data association problem. Amri et al. [17] uses square boxes to model the coverage of motion sensors on the floor plan, and formulate the data association problem within a set-membership estimation framework. Song and Wang [8] introduce a unit disk graph to represent the FoV of each motion sensor, and propose a multi-color particle filter to associate sensor events with the residents.

Additionally, De et al. [18] and Wang et al. [19] propose the idea of mining possible trajectories of smart home residents directly from the recorded sensor events. Each trajectory is a short sequence of sensor events that may be triggered by a resident consecutively. During the tracking phase, various data association hypotheses are created by fitting the mined trajectories to the incoming sensor events. The best hypothesis is chosen so that the average velocity variance is minimized. However, in order to calculate the velocity variance, the distance between any adjacent sensors are required. The algorithm performs better if the number of residents is known during the trajectory mining process.

1.3 SMART HOME FOR AMBIENT ASSISTED LIVING

The Center of Advanced Studies in Artificial Systems (CASAS) group at Washington State University has deployed smart home testbeds and recorded sensor information from the activities in both scripted and unscripted environments. The types of sensors deployed in these testbeds include PIR motion sensors, magnetic door sensors, item presence sensors based on contact pads, power meters, water flow meters, light switches and ambient temperature sensors. For the focus of the multi-resident tracking algorithms presented in this chapter, we focus on the sensor events generated by the PIR motion sensors. In the CASAS smart home testbeds, two kinds of PIR motion sensors are deployed. The first is a downward-facing motion sensor, usually installed on the ceiling, that is sensitive to the resident activities within a 4' $\times$ 4' space underneath it. The downward-facing motion sensors provide an accurate measure of a resident presence at a specific location. As the downward-facing motion sensors deployed in the smart home may not cover the whole space in the house, resulting a lot of resident activities not detected by the smart home, area motion sensors are also installed to fill the gap. The area motion sensors are fitted with a lens so that it can monitor the resident activity within a wide area, and pick up the movement of residents when they are out of the FoV of the downward-facing motion sensors.

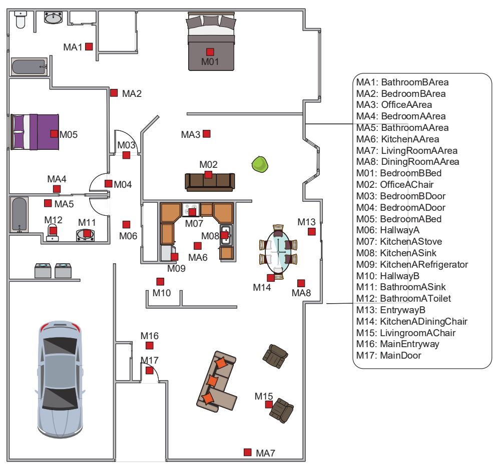

FIGURE 1.1 Floor plan of the smart home TM004 and sensor locations.

There are 25 motion sensors deployed in smart home site TM004. The motion sensors started with "MA" in the ID are fitted with a lens that is responsive to the resident motion in a wider area. The motion sensors started with "M" in the ID are only sensitive to a small calibrated area.

In this chapter, we demonstrate the multi-resident tracking algorithms using a dataset, named TM004, collected in December 2016[1]. There are usually two older adult residents living in the smart home, one of whom has a diagnosis of Parkinson's Disease. However, during the time of the recording, their son and friend may visit and spend the night in the house. The smart home is a two-bedroom apartment, as shown in Figure 1.1, monitored by 25 PIR motion sensors. In Figure 1.1, the sensors with an identifier started with "MA" are area motion sensors, while the others are downward facing.https://www.overleaf.com/project Residents can enter the house from the garage on the bottom left, from the back yard through the door on the right and through the main entrance located at the bottom middle. The TM004 dataset used in this evaluation contains 9 days of annotated data with a total of 98,506 sensor events.

When a PIR sensor detects a resident activity in its FoV, an "ON" message is generated and transmitted to the smart home gateway, where a time tag is added to

[1] The TM004 dataset with annotated sensor events to residents association is available at https://www.stevewang.net/datasets/.

Table 1.1 An example of sensor messages recorded in the TM004 dataset. Each sensor message is a three-tuple consisting of the timestamp, sensor ID and message content. The resident label is provided by annotators. These serve as the ground truth for performance evaluation of multi-resident tracking algorithms

Time Tag	Sensor ID	Message	*Resident*
12/25/2016 15:24:06	LivingRoomAChair	OFF	*R3*
12/25/2016 15:24:08	LivingRoomAArea	OFF	*R1*
12/25/2016 15:24:08	KitchenADiningChair	ON	*R2*
12/25/2016 15:24:08	KitchenAArea	ON	*R2*
12/25/2016 15:24:09	DiningRoomAArea	ON	*R2*
12/25/2016 15:24:10	KitchenADiningChair	OFF	*R2*
12/25/2016 15:24:10	KitchenAArea	OFF	*R2*
12/25/2016 15:24:12	KitchenADiningChair	ON	*R2*
12/25/2016 15:24:12	MainEntryway	ON	*R1*
12/25/2016 15:24:13	KitchenAArea	ON	*R2*
12/25/2016 15:24:13	DiningRoomAArea	OFF	*R2*
12/25/2016 15:24:13	LivingRoomAArea	ON	*R1,R2*
12/25/2016 15:24:13	KitchenADiningChair	OFF	*R2*
12/25/2016 15:24:14	MainEntryway	OFF	*R1*
12/25/2016 15:24:14	KitchenAArea	OFF	*R2*
12/25/2016 15:24:14	LivingRoomAArea	OFF	*R1,R2*
12/25/2016 15:24:22	LivingRoomAArea	ON	*R1,R2*

the message and the message is stored in a central database, and the PIR sensor is in an active state. An "OFF" message follows after the resident motion is no longer present in the FoV of the sensor, and the sensor returns to an inactive state. Thus, in TM004 dataset, each sensor message is a three tuple consisted of the time tag of the message, the sensor identifier, and the message content. Table 1.1 shows a series of sensor messages recorded in the TM004 dataset. As each sensor activation is followed by a deactivation, in this chapter, we use *sensor event* to refer to the subset of sensor messages that contain an "ON" message. The goal of the multi-resident tracking is to associate each sensor event with the residents who activate the sensor.

To evaluate the multi-resident tracking algorithms, external annotators label each sensor event with the identifier for the resident(s) who triggers the sensor message, as shown in the "Resident" column in Table 1.1. The annotators provide the ground truth labels based on the information from raw sensor data and a visualization of sen-

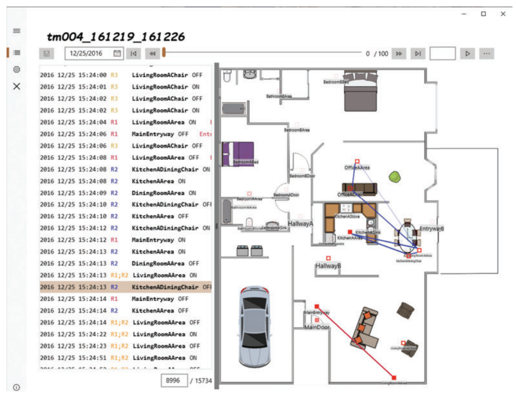

FIGURE 1.2 Screen shot of the ActViz visualization tool.

The ActViz visualization is developed to analyze and annotate the sensor events in multi-resident settings. In the figure, the blue line represents the path of resident "R2" and the red line represents the path of resident "R1". In the mean time, Resident "R3" is sitting in the chair in the living room.

sor observations superimposed on the smart home floor plan, provided by the *ActViz* tool[2]. As shown in Figure 1.2, ActViz maps each sensor events to the smart home floor plan and illustrates the past trajectories of each resident. Thus, the human annotators can examine the movement of each resident along with their previous behavior when annotating each sensor event with the identifiers of corresponding residents. Compared with self-reporting, generating ground truth with external annotators minimizes interruptions to the normal activity routine of the residents, while, at the same time, maintaining a better label consistency [20].

For the purpose of multi-resident tracking, we first extract a "sensor sequence" by focusing on the "ON" message of the PIR motion sensors, as shown in Table 1.2. In single-resident settings, *mutual information* (MI), representing the likelihood that two sensors generating consecutive events, can be estimated using the sensor sequence [20]. Similarly, in the multi-resident environment, sensor pairs with a stronger MI relationship will occur close to each other in the recorded sensor event stream. Hence, the sensor sequence can still provide valuable information about the spatio-temporal relationship between sensors in a multi-resident smart home.

When a sensor is activated, we can take a snapshot of all sensor states in the smart

[2] The *ActViz* tool is a visualization tool developed for annotating and analyzing sensor events of CASAS smart home, especially in multi-resident settings. The tool can be downloaded at https://www.github.com/TinghuiWang/ActViz.git

Table 1.2 Sensor sequence extracted from sensor message shown in Table 1.1

Time Tag	Sensor ID
12/25/2016 15:24:08	KitchenADiningChair
12/25/2016 15:24:08	KitchenAArea
12/25/2016 15:24:09	DiningRoomAArea
12/25/2016 15:24:12	KitchenADiningChair
12/25/2016 15:24:12	MainEntryway
12/25/2016 15:24:13	KitchenAArea
12/25/2016 15:24:13	LivingRoomAArea
12/25/2016 15:24:22	LivingRoomAArea

Table 1.3 Sensor observations, recorded each time a sensor is activated.

Time Tag	Observation
12/25/2016 15:24:08	KitchenADiningChair, MainDoor
12/25/2016 15:24:08	KitchenADiningChair, MainDoor, KitchenAArea
12/25/2016 15:24:09	DiningRoomAArea, KitchenADiningChair, MainDoor, KitchenAArea
12/25/2016 15:24:12	DiningRoomAArea, KitchenADiningChair, MainDoor
12/25/2016 15:24:12	DiningRoomAArea, KitchenADiningChair, MainDoor MainEntryway
12/25/2016 15:24:13	DiningRoomAArea, KitchenADiningChair, MainDoor, KitchenAArea, MainEntryway
12/25/2016 15:24:13	LivingRoomAArea, KitchenADiningChair, MainDoor, KitchenAArea, MainEntryway
12/25/2016 15:24:22	LivingRoomAArea, MainDoor

home. In the snapshot, each active sensor represents an observation of a resident activity. We use term *sensor observations* to describe the set of active sensors in the snapshot. Table 1.3 shows a series of sensor observations extracted from the sensor message in Table 1.1. The relationship between sensor messages, sensor events and sensor observations is illustrated in Figure 1.3. In the graph, each vertical grid line represents the time a sensor in the smart home is activated. The circle represents the sensor observations, and the shaded box represents the time period that a sensor is in active state. According to the annotated labels of residents for each sensor

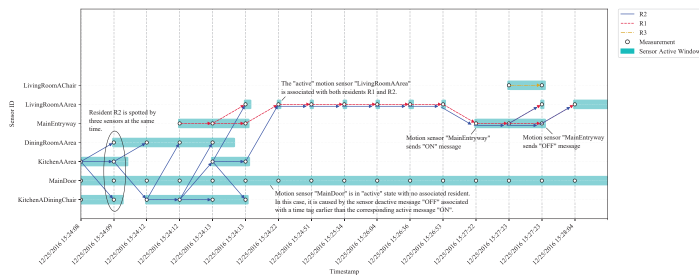

FIGURE 1.3 Association between residents and sensor events in TM004.

The figure shows the relationship among sensor messages, sensor events and sensor observations. The figure is generated using sensor messages recorded in TM004 dataset from the same period as Table 1.1–1.3. The arrows in the graph show the movement of all active residents with respect to sensor observations.

event, the red, blue and yellow arrows show the trajectories of residents R1, R2 and R3, respectively. As shown in Figure 1.3, a resident (R2) may be associated with multiple sensor observations ("DiningRoomAArea", "KitchenAArea" and "KitchA-DiningChair") at the same time, while a sensor observation ("LivingRoomAArea") may be associated with multiple residents (R1 and R2). Moreover, some sensor observations, such as "MainDoor" in the graph, are not associated with any resident. In the context of multi-resident tracking, we use the term *false alarms* or *clutter process* to refer to such sensor observations.

1.4 MULTI-RESIDENT TRACKING IN SMART HOMES

The objective of multi-resident tracking is to find the association between sensor events and residents in the smart home, and, at the same time, estimate the number of active residents. In this section, we introduce three multi-resident tracking algorithms: 1) nearest neighbor with sensor graph (NN-SG); 2) global nearest neighbor with sensor graph (GNN-SG); and 3) sensor-vectorization based multi-resident tracking (sMRT). Both NN-SG and GNN-SG rely on additional information about the sensor adjacency and annotated data to construct the sensor graph. However, in real-life deployment, these information may not be available or inconvenient to obtain. On the contrary, sMRT offers an alternative solution that constructs the resident dynamic model direct from a recorded sensor event stream without any additional information, and tracks multiple residents using Gaussian mixture probability hy-

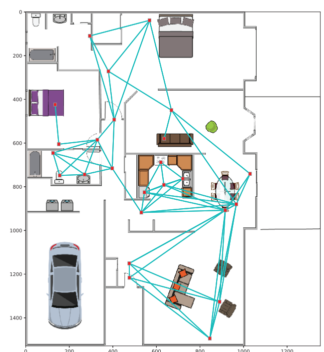

FIGURE 1.4 Sensor adjacency in the smart home TM004.

The adjacent sensors in smart home TM004 are joined with a blue line. Two sensors are adjacent if a resident can activate the sensors consecutively without triggering any other sensor in the smart home.

pothesis density filter and a customized track maintenance algorithm.

1.4.1 NN-SG: NEAREST NEIGHBOR WITH SENSOR GRAPH

The NN-SG algorithm is an extension of GR/ED algorithm proposed in earlier work by Crandall and Cook [9]. A sensor graph is a bidirectional graph where the vertexes of the graph are mapped to the sensors in the smart home. If the movement of a resident can trigger sensor s_i and sensor s_j consecutively without activating any other sensor in the smart home, sensor s_i and sensor s_j are adjacent in the sensor graph. For example, Figure 1.4 illustrates the adjacency between PIR motion sensors deployed in smart home TM004.

The weight on the directional edge from sensor s_i to sensor s_j in the sensor graph represents the conditional probability of a resident activating sensor j after sensor i, $Pr(s_j|s_i)$. In another word, the sensor graph is equivalent to a Markov chain, where the states of the Markov chain correspond to the nodes in the sensor graph. The weight on the directional edges of the sensor graph forms the transition matrix $\mathbf{P}$ of the Markov chain, with $p_{ij} = Pr(s_j|s_i)$. If sensor s_i and s_j are not adjacent, $p_{ij} = 0$. Thus, given a recorded sensor sequence with annotated labels for resident association, the values in the transition matrix can be estimated by maximizing the likelihood of generating the sensor sequence. For instance, based on the association labels provided by the annotator in TM004 dataset, the estimated transition matrix is shown in Figure 1.5.

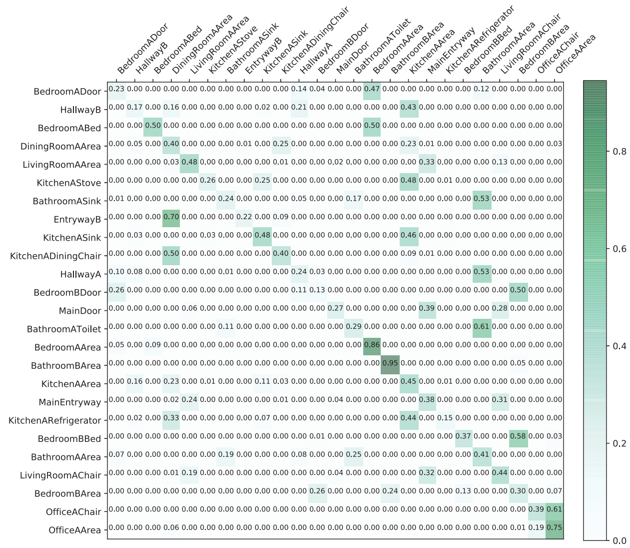

FIGURE 1.5 Transition matrix of the sensor graph.

Each entry in the figure represents the probability of a resident moving from the sensor in the row to the sensor in the column. For example, the 0.47 in the top row represents the conditional probability of resident activating sensor "BedroomAArea" after sensor "BedroomADoor" is 0.47.

With the sensor graph, NN-SG uses the nearest neighbor algorithm to associate sensor events with existing residents in the smart homes. However, in order to initiate a new track for a resident who just enter the house, or remove an old track when the corresponding resident leaves the house or becomes inactive, the following set of rules, originally developed in prior work of Crandall and Cook [9], are adopted.

Rule of target death An existing target (resident) is assumed to have left the house or become "inactive" if the target has not been detected by any sensors for a period of 50 sensor events (the parameter is suggested in GR/ED).

Rule of target birth If a sensor event is not found associated with any existing targets (residents), a new target will be formed and associated with the sensor event.

Whenever a new sensor event arrives, the NN-sg method first search through the existing active tracks. If an existing track is previously spotted by an adjacent sensor, the track is associated with the sensor event. However, if multiple existing tracks are found, the one with the highest likelihood of activating the current sensor is associated with the sensor event. When no existing track is found previously spotted by an adjacent sensor, according to the rule of target birth, a new target is spawn. NN-SG then check each existing target against the rule of target death and remove the dead target from the list before moving on to the next sensor event.

1.4.2 GNN-SG: GLOBAL NEAREST NEIGHBOR WITH SENSOR GRAPH

GNN-SG contrasts with NN-SG by associating targets with sensor observations. At each time step, GNN-SG generates a list of all possible one-to-one associations between the sensor observations (all active sensors) and existing residents. A score is assigned to each association hypothesis by accumulating the probability of each existing track to the new sensor location according to the sensor graph. The hypothesis with best score is selected, and any sensor observation that is not associated with any resident is considered the start of a new track and issued with a new target identifier. The hypothesis selection process is equivalent to the binary assignment problem, which can be solved efficiently using the Hungarian algorithm [21].

1.4.3 SMRT: MULTI-RESIDENT TRACKING WITH SENSOR VECTORIZATION

sMRT formulates the multi-resident tracking problem as a sequential Bayes estimation (or filtering) problem in the framework of finite set statistics (FISST) [22]. The state of each resident, denote $\mathbf{x}$, is a random vector that belongs to a state space $\mathcal{X}$. Thus, the states of all active residents in the smart home can be represented as a random finite set (RFS) $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \in \mathcal{F}(\mathcal{X})$, where $\mathcal{F}(\mathcal{X})$ is the collection of all finite subsets of the state space $\mathcal{X}$. Each element $\mathbf{x}_i (1 \leq i \leq n)$ of the RFS X is a state vector of an active resident. The total number of active residents in the smart home, n (i.e., the cardinality $|X|$), is a random variable defined on $\mathbb{Z}_0^+$. Given a sequence of sensor events, sMRT calculates a Bayes optimal probability density, $f(X_k)$, of the RFS X_k at time step k. The number of active residents, or the cardinality of the RFS X_k, is simultaneously derived.

To bridge the input (a series of sensor observations) and the output (the association between sensor events and identified tracks), sMRT is composed of two phases: a learning phase and a tracking phase. During the learning phase, we first map the PIR motion sensors in the smart home into a latent space based on the recorded sensor sequence. We further hypothesize that the dynamic model of the resident movement in the smart home can be represented by a constant velocity model of a point target maneuvering in the space. During the tracking phase, we used a Gaussian mixture probability hypothesis density (GM-PHD) filter in combination with a track maintenance algorithm to derive the association between sensor observations

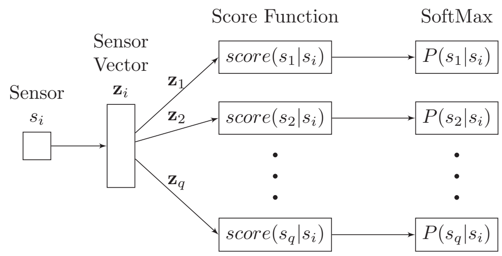

FIGURE 1.6 The generative model of sensor vectorization.

The generative model of sensor vectorization predicts the likelihood of a resident activates sensor s_i and sensor s_j consecutively based on their vector representations $\mathbf{z}_i$ and $\mathbf{z}_j$ in the measurement space.

and identified targets (residents) in the smart home. In contrast with the NN-SG and GNN-SG methods, sMRT constructs the dynamic model solely based on a series of recorded sensor data, without any additional information that may raise privacy concerns or is impractical to acquire for real homes.

1.4.3.1 Sensor Vectorization

In a smart home with q PIR motion sensors, denoted $s_1, s_2, \dots, s_q$, the training phase of sMRT begins by mapping each sensor s_i as a vector $\mathbf{z}_i$ in a m dimensional latent space $\mathcal{Z}$. As each sensor observation serves as a measurement of the resident activity, we refer to this latent space as the measurement space. In order to fit the resident movement in the smart home with a constant velocity model in the measurement space, the vector representations are created by mining the spatio-temporal relationship exhibited in the recorded sensor sequence. Intuitively, the higher the conditional probability is of a resident activating sensor s_i and sensor s_j consecutively in the sensor stream, the closer are the corresponding vector representations $\mathbf{z}_i$ and $\mathbf{z}_j$ in the measurement space. In a multi-resident scenario, the recorded sensor sequence is a time-ordered collection of the active sensor messages associated with all residents in the smart home, possibly moving through different parts of the home. As a result, adjacent sensors are not necessarily next to each other in the sensor event sequence. However, they are more likely to show up within c sensor messages apart, where c is an integer that can be selected based on the expected number of smart home residents. Thus, we construct a generative model that predicts the probability of two sensors being adjacent parameterized by their vector representations in measurement space. This probability needs to fit the sensor pair's co-occurrence observed in the recorded sensor sequence within a window of c sensor messages.

Formally, given a sensor sequence containing M sensor messages, $s^{(1)}, \dots, s^{(M)}$,

$s^{(i)}$ is the corresponding sensor ID, we generate a training set where each sensor pair is observed within a window of c sensor messages in the sensor sequence, as shown in Eq. (1.1).

$$\textit{training set} = \{(s^{(i)}, s^{(j)}) | 0 < j - i \leq c\} \tag{1.1}$$

We construct a generative model (as shown in Figure 1.6) that predicts the probability of a sensor pair s_i and s_j being adjacent in the smart home, denoted as $P(s_i|s_j) = P(s_j|s_i)$. With the probability $P(s_i|s_j)$ as a function of the corresponding vector representation $\mathbf{z}_i$ and $\mathbf{z}_j$, the vector representations of all sensors in the measurement space can be trained by maximizing the average log likelihood $\mathcal{L}$ of the sensor pairs observed in the training set, as shown in Eq. (1.2).

$$\mathcal{L} = \frac{1}{M} \sum_{i=1}^{M} \sum_{0<j-i\leq c} \log P(s^{(j)}|s^{(i)}) \tag{1.2}$$

The probability of sensor s_i being adjacent to sensor s_j can be defined using a SoftMax function based on a score assigned to them, as shown in Eq. (1.3).

$$P(s_j|s_i) = \frac{\exp(\text{score}(s_j|s_i))}{\sum_{k=1}^{q} \exp(\text{score}(s_k|s_i))} \tag{1.3}$$

The score value $\text{score}(s_j|s_i)$ needs to be larger when the distance between the corresponding vectors is smaller. We use a dot product as the similarity measure that defines the score function, as shown in Eq. (1.4).

$$\text{score}(s_j|s_i) = \text{score}(s_i|s_j) = \mathbf{z}_i \cdot \mathbf{z}_j^T \tag{1.4}$$

In a smart home containing a small number of sensors, the vector representations of sensors in the measurement space can be learned directly using SoftMax cross-entropy loss. To reduce the large computational cost of directly learning vector representations for a large number of sensors, noise contrast estimation (NCE) [23] is employed.

1.4.3.2 Linear Gaussian Dynamic Model

With each sensor in the smart home mapped into the measurement space, we use a constant velocity model of a point target maneuvering in the measurement space to approximate the movement of each resident in the smart home. The state vector of each resident is a $(2m+1) \times 1$ vector $\mathbf{x} = \begin{bmatrix} x^T & v^T & r \end{bmatrix}^T$, where x is an $m \times 1$ vector representing the location of the resident in space $\mathcal{Z}$, v is an $m \times 1$ vector representing the velocity of the resident, and r is an integer representing the resident identifier or the track identifier generated by sMRT. According to the constant velocity assumption, the resident state $\mathbf{x}$ at the next time step can be estimated based on the resident's current state, $\mathbf{x}'$, as shown in Eq. (1.5). Here, $\mathbf{F}$ represents the linear motion multiplier, $\mathbf{G}$ represents the linear error multiplier, and $\mathbf{w}$ represents the velocity error.

$$\mathbf{x} = \mathbf{F} \cdot \mathbf{x}' + \mathbf{G} \cdot \mathbf{w} \tag{1.5}$$

If $\mathbf{w}$ can be modeled using a Gaussian distribution, the probability distribution of the resident state at the next time step can be expressed using a linear Gaussian model as in Eq. (1.6). In the equation, $\mathbf{Q}$ is the resulting covariance matrix.

$$f(\mathbf{x}|\mathbf{x}') = \mathcal{N}(\mathbf{x}; \mathbf{F}\mathbf{x}', \mathbf{Q}) \tag{1.6}$$

In the smart home, motion sensors will be activated by the resident activities within an area defined by the FoV of the sensor, the sensor observations (represented by the corresponding sensor vectors) offer a noisy measurement of true resident states. If we assume that such measurement errors can be modeled as a Gaussian distribution with zero mean and a covariance matrix $\mathbf{R}$, the relationship between a sensor observation $\mathbf{z}$ and the state vector $\mathbf{x}$ of the resident can also be represented using a linear Gaussian model as shown in Eq. (1.7) with linear multiplier $\mathbf{H}$.

$$f(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \mathbf{H} \cdot \mathbf{x}, \mathbf{R}) \tag{1.7}$$

The hypothesis that the resident movement in the smart home can be fitted with a constant velocity model in the measurement space is a strong assumption, and may not hold true in real life. However, with the help of the GM-PHD filter and track maintenance algorithm proposed in the following sections, deviations between the reality and the assumption can be captured by the Gaussian noise in the dynamic model and the measurement model shown in Eq. (1.6) and Eq. (1.7). Thus, the GM-PHD filter can correct these errors based on the sensor observations obtained at each step.

1.4.3.3 GM-PHD Filter

Provided with the vector representation of each sensor, the sensor observations extracted from the sensor event stream are translated into a set of vectors in the measurement. At time step k, we define an *observation set* $Z_k = \{\mathbf{z}_1, \ldots, \mathbf{z}_{n_z}\}$, where n_z is the number of active sensors and each element $\mathbf{z}_i$ is the vector representation of the corresponding sensor. Among these n_z sensor observations, some are accurate measurements of active residents and some are false alarms (or clutter) due to communication errors or sensor failures. Alternatively, some residents may still be at home but may not be currently detected by the sensors. Thus, in addition of mapping each sensor observation with the existing targets (residents) identified in the previous steps, we also need to consider the possibilities of a new resident entering the home, an existing resident leaving the home, residents not being detected, sensor

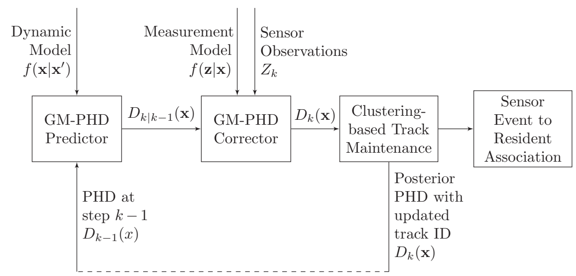

FIGURE 1.7 The sMRT tracking phase.

During the tracking phase, GM-PHD predictor updates the PHD of resident states according to the dynamic model, and GM-PHD corrector corrects the prediction with the measurement model and sensor observations. Given the corrected PHD of resident states, track maintenance algorithm updates the track identifiers in the PHD and the association between sensor observations and identified tracks are derived.

observations not being associated with any resident, and one-to-many or many-to-one associations between sensor observations and residents.

To model all of these possibilities, we use a Gaussian mixture probability density (GM-PHD) filter [24] that propagates the first-order moment of the multi-target probability density, or the probability hypothesis density (PHD), based on the dynamic and measurement models constructed during the learning phase. Additionally, we propose clustering-based track maintenance to associate the PHD predicted by the GM-PHD filter with resident identifiers to detect new residents while maintaining the traces of existing residents. Finally, each sensor observation, represented as a vector in the measurement space, is associated with the resident that is most likely to generate the observation. The steps of the tracking phase are illustrated in Figure 1.7.

The GM-PHD filter is composed of a predictor and a corrector. Given the PHD of multiple residents at time step $k-1$, $D_{k-1}(\mathbf{x})$, the predictor estimates the multi-resident PHD at time step k, $D_{k|k-1}(\mathbf{x})$, based on the linear Gaussian dynamic model in (1.6). The corrector then refines the predicted PHD, $D_{k|k-1}(\mathbf{x})$, based on the measurement model and sensor observations, Z_k. The output of the corrector is the Bayes optimal estimation of the posterior multi-resident PHD at time step k, $D_k(\mathbf{x})$, which can be used to associate sensor events with residents in the smart home. If the multi-resident PHD at time step $k-1$, $D_{k-1}(\mathbf{x})$, is in the form of a Gaussian mixture, and the dynamic model and the measurement model are both linear Gaussian, the resulting posterior multi-resident PHD, $D_k(\mathbf{x})$, is guaranteed to be in the form of a Gaussian mixture, as shown in (1.8), where J_k is the number of Gaussian components in the mixture and $w_k^{(i)}$, $\mathbf{m}_k^{(i)}$ and $\mathbf{P}_k^{(i)}$ are the weight, mean vector and covariance matrix of the i^{th} Gaussian component, respectively.

$$D_k(\mathbf{x}) = \sum_{i=1}^{J_k} w_k^{(i)} \mathcal{N}(\mathbf{x}; \mathbf{m}_k^{(i)}, \mathbf{P}_k^{(i)}) \tag{1.8}$$

1.4.3.4 Track Maintenance and Data Association

Given the posterior PHD at time step k, we propose a clustering-based track maintenance algorithm that estimates the state of each resident, assigns identifiers to the newly-identified residents, and associates sensor observations with each resident based on the state probability distribution of each identified resident. According to the definition of PHD, the expected number of residents in the smart home can be calculated by integrating the PHD over the entire state space, as shown in Eq. (1.9).

$$N_k = \int \sum_{i=1}^{J_k} w_k^{(i)} \mathcal{N}(\mathbf{x}; \mathbf{m}_k^{(i)}, \mathbf{P}_k^{(i)}) d\mathbf{x} = \sum_{i=1}^{J_k} w_k^{(i)} \tag{1.9}$$

We first assume that, at any time step, there is at most one newly-detected resident. Thus, during the predictor step, we can assign a new resident identifier to the resident ID field of the Gaussian mean state vectors for the target birth PHD. Given the measurement model and the dynamic model, the resident identifier in the mean vector of each Gaussian component will remain unchanged while the Gaussian components are propagated in time through the GM-PHD filter. By grouping together the Gaussian components that share the same resident identifier in the mean vector, the state probability distribution of each resident can be derived.

We now consider the case that multiple residents, $R_1^{(k)} \dots R_{n_r^{(k)}}^{(k)}$, enter the smart home at time k. As we assign a single resident identifier, $r^{(k)}$, to all Gaussian components in the target birth PHD, the Gaussian components of the PHD, representing the states of all residents entering the smart home, share the same resident identifier $r^{(k)}$. As the residents move through time, the cardinality of the PHD will eventually approximate the actual number of residents, $N^{(k)}$, who enter the home. As a result, when tracking each resident $R_i^{(k)}$, the Gaussian components representing the PHD of those $N^{(k)}$ residents need to be separated into $N^{(k)}$ clusters with a unique resident identifier assigned to the Gaussian components for each cluster.

In sMRT, we introduce a clustering-based track maintenance algorithm that monitors the integral of the PHD associated with each resident identifier. The track maintenance algorithm is an iterative six-step process as follows.

1. Given the PHD with resident identifier r in the form of a Gaussian mixture as shown in Eq. (1.10), calculate the number of expected residents $N'_{k,r}$ as shown in Eq. (1.11).

$$D_{k,r}(\mathbf{x}) = \sum_{i=1}^{J_{k,r}} w_{k,r}^{(i)} \mathcal{N}\left(\mathbf{x}; \mathbf{m}_{k,r}^{(i)}, \mathbf{P}_{k,r}^{(i)}\right) \tag{1.10}$$

$$N'_{k,r} = \lceil N_{k,r} - 0.5 \rceil = \left\lceil \sum_{i=1}^{J_{k,r}} w_{k,r}^{(i)} - 0.5 \right\rceil \tag{1.11}$$

2. Initialize the center of $N'_{k,r}$ clusters randomly as $\alpha_1, \ldots, \alpha_{N'_{k,r}}$.
3. For each cluster, find the Gaussian components in $D_{k,r}(\mathbf{x})$ with the smallest distance between the mean of the Gaussian component and the center of the corresponding cluster. Assign those Gaussian components to the cluster so that the summation of the weights of all those Gaussian components does not exceed $N_{k,r}/N'_{k,r}$. If there are Gaussian components left not assigned to any cluster, assign each of these to the nearest cluster determined by the distance between the center of the cluster and the mean of the Gaussian component.
4. Update the cluster center α_j to be the weighted mean of all Gaussian components assigned to the cluster, as shown in Eq. (1.12).

$$\alpha_j = \frac{1}{\sum_{i=1}^{J_{k,r,j}} w_{k,r,j}^{(i)}} \sum_{i=1}^{J_{k,r,j}} w_{k,r,j}^{(i)} \mathbf{m}_{k,r,j}^{(i)} \tag{1.12}$$

   In Eq. (1.12), $J_{k,r,j}$ represents the number of Gaussian components assigned to cluster j. The $\mathbf{w}_{k,r,j}^{(i)}$, $\mathbf{m}_{k,r,j}^{(i)}$ terms represent the weight and mean of those Gaussian components.
5. Repeat steps 3 and 4 until there are no further changes to the association between Gaussian components and clusters, or a maximum number of iterations is reached.
6. With the Gaussian components segregated into $N'_{k,r}$ clusters, a new resident identifier is assigned to each cluster and is inserted into the resident ID field in the mean vector of each Gaussian component assigned to that cluster.

Finally, each sensor observation $\mathbf{z}_i \in Z_k$ is associated with the resident ID r so that the likelihood of producing the sensor observation $\mathbf{z}_i$ is maximized, as shown in Eq. (1.14).

$$r = \operatorname{argmax}_r \int f(\mathbf{z}_i|\mathbf{x}) \sum_{i=1}^{J_{k,r}} w_{k,r}^{(i)} \mathcal{N}\left(\mathbf{x}; \mathbf{m}_{k,r}^{(i)}, \mathbf{P}_{k,r}^{(i)}\right) d\mathbf{x} \tag{1.13}$$

$$= \operatorname{argmax}_r \sum_{i=1}^{J_{k,r}} w_{k,r}^{(i)} \mathcal{N}\left(\mathbf{z}; \mathbf{H}\mathbf{m}_{k,r}^{(i)}, \mathbf{R} + \mathbf{H}\mathbf{P}_{k,r}^{(i)}\mathbf{H}^T\right) \tag{1.14}$$

1.5 PERFORMANCE METRICS

In this section, we introduce the three sets of performance metrics to evaluate the MRT algorithms presented in this chapter. First, we evaluate the output of multi-resident tracking algorithms in the framework of multi-class classification. We use accuracy score, Hamming loss, precision, recall, and F1-score to compare the performance of each tracking algorithm against the ground truth. This set of metrics is commonly used in past research, especially when the number of residents in the smart home is assumed to be fixed. In addition, we also want to evaluate how well the tracking algorithm can estimate the number of active residents in the smart home. Thus, the second metric we use is the average error in the number of active residents estimated by the residents. Finally, we adapt the multi-object tracking accuracy (MOTA), commonly used for multi-object tracking in video surveillance applications, to the multi-resident tracking problem, and propose the multi-resident tracking accuracy (MRTA). By focusing on the error categories, including target misses, false positives, and target identifier mismatch errors, MRTA provides additional statistics and insights to debug and improve the algorithm.

1.5.1 TRACKING AS MULTI-CLASS CLASSIFICATION

The goal of a multi-resident tracking algorithm is to associate each sensor events with the residents in the smart home. If the number of the residents in the smart home is fixed or the maximum number of the residents of a dataset is given, we can treat the output of the multi-resident tracking algorithm as classifying each sensor events into multiple classes, each of which represents a resident in the smart home. Thus, common performance measures for multi-class classification problem, such as accuracy score, Hamming loss, precision, recall, and F1-score can be used to compare the performance between tracking results.

Before computing the metrics, the target identifiers generated by the tracking algorithm need to be mapped to the resident identifiers annotated in the ground truth. To create such correspondence, we first group the sensor events associated with each target identifier. We then find the resident identifier who associates to most of those sensor events according to the ground truth. Thus, a one-to-one mapping between target identifiers of the tracking algorithm and resident identifiers in the ground truth is formed. Based on the mapping, each sensor event is updated with resident identifiers labels, and the multi-class classification metrics can be calculated.

We define *association accuracy* as the fraction of total sensor events, D, in which the ground truth $Y^{(i)}$ equals the set of predicted resident IDs $\hat{Y}^{(i)}$, as shown in Eq. (1.15). Resident identifiers include the empty set (no resident) or a set of identifiers for one or more residents.

$$accuracy = \frac{1}{D} \sum_{i=1}^{D} \mathbf{1}(Y^{(i)} = \hat{Y}^{(i)}) \tag{1.15}$$

Hamming loss, on the other hand, gives credits to partial matches between $Y^{(i)}$ and $\hat{Y}^{(i)}$. The definition of Hamming loss is shown in Eq. (1.16). In Eq. (1.16), N_R represents the total number of residents in the dataset.

$$hamming\ loss = \frac{1}{D} \cdot \frac{1}{N_R} \sum_{i=1}^{D} \sum_{j=1}^{N_R} \mathbf{1}\left(y_j^{(i)} = \hat{y}_j^{(i)}\right) \tag{1.16}$$

Moreover, if we focus on each resident that is annotated in the ground truth, we can also view sensor event to resident association as a binary classification problem. The two classes are events that are associated with a particular resident (+) and events not associated with that resident (-). In this approach, we can measure the precision, recall, and F1-score for each resident.

1.5.2 ERROR IN ESTIMATED NUMBER OF ACTIVE RESIDENTS

However, as the multi-class classification metrics is computed with a constant number of total classes for each sensor events, it fails to address the scenario that the number of active residents in the smart home may vary from time to time. For a tracking algorithm to work in a real-life environment, estimation of the number of active residents in the house can provide valuable information. Every target identifier generated by the tracking algorithms represents a potential resident. Thus, we calculate the number of active target identifiers at each time step and compute the error against the number of active residents annotated in the ground truth. In earlier multi-resident tracking research, a resident is considered to be inactive if the resident has not been detected by any sensors for over 100 seconds, or 50 consecutive sensor events on average [9]. This rule is applied to both the ground truth and the target identifiers generated by NN-SG and GNN-SG method. In case of sMRT, the likelihood of a resident being at any time step, can be calculated by integrating the corresponding PHD, as shown in Eq. (1.11). If the likelihood is greater than 0.5, we consider the target identifier to be active.

1.5.3 MULTI-RESIDENT TRACKING ACCURACY (MRTA)

Past research on multi-object tracking in computer vision applications has proposed MOTA metric to extract the accuracy aspect of the system output. The MOTA metric focuses on the potential errors that may occur in the output of a tracking system, including target miss, false positive target identification, and target identifier mismatch. In computer vision applications, the association between the target identified by the tracking system and the ground truth can be established by the size of overlap area or the physical distance in a video frame. However, in multi-resident tracking applications, the observation and target identifier are discrete, and a one-to-one association may be violated. Thus, we propose MRTA by adapting MOTA in the context of multi-resident tracking.

As with the computation of multi-class classification metrics, we first establish the correspondence between target identifier generated by tracking algorithms and the resident identifiers in the ground truth. We then classify the errors between the tracking algorithm outputs and the ground truth labels into the following three categories: misses, false positives, and mismatches.

Misses If a sensor is associated with a resident while in the tracking algorithm, but there is no track identified that is mapped to that resident, the association is counted as a miss.

False Positives If a sensor event is associated with a resident and there are multiple tracks generated by the tracking algorithm which all map to the same resident, the association is considered a false positive. Similarly, if a track identified by the tracking algorithm is associated with a resident that is not linked to the sensor event according to ground truth, this association is considered a false positive.

Mismatch If a resident is still "active" according to ground truth, while the track identifier changes in the algorithm output, the corresponding associations are considered mismatches.

The MRTA score can be calculated according to Eq. (1.17).

$$\text{MRTA} = 1 - \frac{N_{misses} + N_{fp} + N_{mismatch}}{N_{association}} \tag{1.17}$$

In the equation, N_{misses} is the number of target misses, N_{fp} is the number of false positives and $N_{mismatch}$ is the number of target identifier mismatches. The $N_{association}$ in the denominator represents the total number of identified event to resident associations that were annotated in the ground truth. For example, if a sensor event is associated with two residents, the number of ground truth associations is also two.

The accuracy score and Hamming loss only focus on the correctness of the association hypothesis generated by the tracking algorithms. However, when the tracking algorithm generates multiple target identifiers corresponding to the same resident in the ground truth at the same time, the accuracy score and Hamming loss does not penalize those errors. On the contrary, MRTA counts those extra target identifiers as false positives.

1.6 EXPERIMENTS AND DISCUSSION

In the experiment, we evaluate the performance of the three multi-resident tracking methods, NN-SG, GNN-SG and sMRT presented in this chapter with TM004 dataset introduced in Section 1.3. We require that each valid track be composed of at least three sensor events. In earlier activity recognition research, the shortest detectable activities contained at least three events (the "enter home" and "leave home" activities). Thus, if a target identifier in the output of a tracking algorithm is associated with fewer than three sensor events, we consider those sensor events are false alarms

Table 1.4 Accuracy score and Hamming loss of NN-SG, GNN-SG and sMRT. The best performance values are shown in bold. The best performance values that are statistically significant ($p < 0.5$) are marked with an asterisk.

Methods	sMRT	NN-SG	GNN-SG
Accuracy	0.80	0.80	**0.83***
Hamming loss	0.08	0.09	**0.07**
# Tracks	2834	569	1441
# Sensor Events	51358		

Table 1.5 Performance of NN-SG, GNN-SG and sMRT measured based on binary classification accuracy on a per-resident basis. The number in parentheses after each resident identifier refers to the number of associated sensor events in the ground truth.

Metrics	Precision			Recall			F1-Score		
Methods	sMRT	NN-SG	GNN-SG	sMRT	NN-SG	GNN-SG	sMRT	NN-SG	GNN-SG
R1 (32,272)	0.94	0.89	0.92	0.84	0.85	0.91	0.89	0.87	0.91
R2 (17,873)	0.86	0.80	0.86	0.76	0.73	0.78	0.81	0.77	0.82
R3 (1,202)	0.85	0.77	0.76	0.71	0.79	0.85	0.77	0.78	0.80
R4 (11)	0.00	0.00	0.67	0.00	0.00	0.73	0.00	0.00	0.70
Micro Avg.	0.91	0.85	0.90	0.81	0.81	0.86	0.86	0.83	0.88
Macro Avg.	0.66	0.61	0.80	0.58	0.59	0.82	0.62	0.60	0.81

and discard the target identifier. Table 1.4 shows the multi-classification accuracy score and Hamming loss of NN-SG, GNN-SG, and sMRT.

According to the multi-classification accuracy, sMRT ties with NN-SG with an accuracy of 0.80, while GNN-SG scores the best of 0.83. On the Hamming loss, sMRT scores 0.08, 0.01 better than the NN-SG method. The GNN-SG performs the best with a Hamming loss of 0.07. Based on the above result, GNN-SG achieved the best performance in terms of multi-class classification metrics and identified 93% of sensor event to resident associations correctly. It is also worth noting that both NN-SG and GNN-SG requires sensor adjacency in the smart home as a prerequisite, while the sMRT achieved only 1% worse than GNN-SG and 1% better than NN-SG without such information.

When we break down the output of the tracking algorithms on a per-resident basis, the precision, recall, and f1-scores achieved by all the methods are presented in Table 1.5. While the macro averages are commonly reported when the classes are imbalanced, we are also interested in results on a per-datapoint basis. Thus, both

Table 1.6 Average error in estimated number of active residents. The best performance values are shown in bold. The best performance values that are statistically significant ($p < 0.5$) are marked with an asterisk.

Methods	sMRT	NN-SG	GNN-SG
Average Error	0.59	**0.41***	1.27

Table 1.7 MRTA performance of NN-SG, GNN-SG and sMRT. The best performance values are shown in bold. The best performance values that are statistically significant ($p < 0.5$) are marked with an asterisk.

Methods	sMRT	NN-SG	GNN-SG
MRTA	0.47	**0.69***	0.41
Misses	9,879	7,435	**7,024**
False positives	12,602	**5,230***	10,334
Mismatches	4,371	**3,331**	13,078
Total Associations	51,358		

micro averages and macro averages are provided.

In the dataset, residents R1 and R2 are present and active most of the time, with 32,272 and 17,873 sensor events associated with them, respectively. Resident R3 and R4 are likely visitors, associated with 1,202 and 11 sensor events, respectively. According to Table 1.5, sMRT achieves a better precision score for residents R1, R2 and R3, while GNN-SG acheives a better recall across the board. Both sMRT and MRT failed to identify resident R4 due to the limited presence of the resident in the dataset.

Table 1.6 shows the average errors in the estimated number of active residents for the NN-SG, GNN-SG and sMRT algorithms. Both the NN-SG and sMRT methods, on average, are accurate for the estimation of the number of active residents, as the errors of both methods are below 1. On the contrary, GNN-SG generates a higher number of valid target identifiers and fails to estimate the number of active residents as accurately as other methods.

The MRTA performances of NN-SG, GNN-SG and sMRT are shown in Table 1.7. NN-SG achieves the best MRTA score of 0.68, with sMRT trailing at 0.47 and GNN-SG at 0.40. When we break down the tracking errors into misses, false positives, and mismatches, we find that NN-SG has the lowest false positives and mismatches. GNN-SG has the lowest number of target misses, but exhibits extremely high counts of false positives and track ID mismatches. However, the result of sMRT shows high number of misses and false positives compared with NN-SG and GNN-SG, but the algorithm achieves a MRTA of 0.47, higher than GNN-SG.

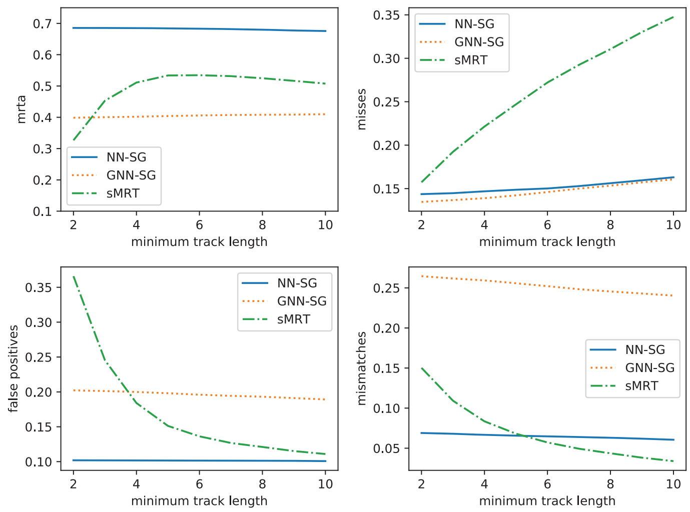

FIGURE 1.8 MRTA performance versus the minimum length of sensor events.

By varying the number of sensor events a valid target identifier should be associated with, we plot the MRTA score (top left), ratio of target misses $N_{misses}/N_{associations}$ (top right), ratio of false positives $N_{fp}/N_{associations}$ (bottom left), and ratio of target identifier mismatches $N_{mismatches}/N_{associations}$ (bottom right).

During the calculation of the results shown in Table 1.7, we require that each valid target identifier be associated with at least three sensor events. The minimum number of sensor events is determined heuristically. Figure 1.8 shows the impact of the minimum number of sensor events on the MRTA metrics and the number of different tracking errors. If the minimum number of sensor events associated with a valid target increases, an increase in target misses is observed among all three algorithms, with sMRT increasing most rapidly. However, if we require each valid target is associated with more sensor events, the false positives and target mismatches of sMRT drops rapidly, and the MRTA score of sMRT may reach 0.56. The MRTA scores of NN-SG and GNN-SG are more resilient to such changes.

1.7 CONCLUSIONS

In this work, we introduce three approaches to multi-resident tracking algorithms in AAL environments using PIR motion sensors. We also introduce novel evaluation mechanisms to determine the effectiveness of alternative techniques. All three of the described algorithms can handle cases with varying number of residents in the environment. However, NN-SG and GNN-SG rely on sensor locations and environment floor plans to determine sensor adjacency, while sMRT solves the multi-resident data association problem by mining the spatio-temporal relationship of sensors directly from unannotated sensor data without additional information that may raise privacy concerns or impractical to acquire. We evaluate the performance of all the multi-resident tracking algorithms using a smart home dataset recorded in real-life settings with human-annotated association between sensor events and residents. The performance is presented using multi-class classification metrics, average error in the estimation of the number of active residents, and the MRTA metric that we proposed.

According to the results, GNN-SG achieves the best accuracy score and Hamming loss. However, due to the extremely high number of false positive errors and target identifier mismatch errors, both of which are not penalized in the multi-class classification metrics, GNN-SG is the weakest based on MRTA metrics. On the contrary, NN-SG achieves the best MRTA score, while sMRT comes in second, beating GNN-SG by 0.06. NN-SG and sMRT also achieve the top two scores in estimating the number of active residents in the smart home. Considering that sMRT solves the multi-resident tracking without additional information, which NN-SG and GNN-SG do require, the result shows that sMRT, as an initiative for unsupervised multi-resident tracking, is capable of associating sensor events with residents in the real-life settings. Continued research in finding an unsupervised multi-resident tracking solution could help AAL technology to scale to multi-resident homes, thus providing practical benefit to individuals and families needing activity monitoring and activity-aware services.

Though sMRT proposed in this work is an initiative towards multi-resident tracking based solely on unannotated sensor events, the experiment results prompt many possibilities for future improvements. First, the constant velocity model of residents maneuvering in the measurement space is a strong constraint. However, this constraint could potentially be relaxed, since the tracking phase of sMRT works with any linear Gaussian dynamic model. In order to derive the parameters of such linear Gaussian dynamic models, one possible research direction is to take advantage of the generative nature of the sMRT tracking phase.

According to the MRTA metrics, sMRT experiences high misses and false positives. The algorithm also experiences a steep increase in target misses when the minimum valid-track length increases. The result shows that sMRT has difficulties in maintaining a track for a long period of time. Future research in resident identification methods could improve the time-continuity of tracks identified by the tracking algorithm.

Last but not least, the locations and position of sensors in the smart environment may also affect the tracking accuracy in a multi-resident settings. For example, a higher density of sensors in the environment may create more overlaps between sensors, resulting in an increase of the cases where a resident is associated with multiple sensor observations. However, a lower density of sensor deployment may result in a resident remaining undetected for a longer period of time, leading to the resident not being correctly identified by the tracking algorithm. Thus, evaluating the design of an AAL environment and the deployment of sensors is another valuable research direction, especially for smart environments inhabited with multiple residents.

ACKNOWLEDGMENTS

he authors would like to thank Brian Thomas and Aaron Crandall for their assistance in collecting smart home sensor data and Sue Nelson for her assistance in providing ground truth labels for the smart home sensor data. This material is based upon work supported by the National Science Foundation under Grant No. 1543656.

REFERENCE

1. United Nations, Department of Economic and Social Affairs, Population Division, World population prospects 2019: Data booklet. 2019.
2. Monekosso D, Florez-Revuelta F, Remagnino P, Ambient assisted living [guest editors' introduction]. IEEE Intelligent Systems 2015; 30(04):2–6. doi:10.1109/MIS.2015.63.
3. Benmansour A, Bouchachia A, Feham M, Multioccupant activity recognition in pervasive smart home environments. ACM Computing Surveys 2015; 48(3):34:1–34:36.
4. Rashidi P, Mihailidis A, A survey on ambient-assisted living tools for older adults. IEEE Journal of Biomedical and Health Informatics 2013; 17(3):579–590.
5. Lu G, Yan Y, Ren L, Saponaro P, Sebe N, Kambhamettu C, Where am i in the dark: Exploring active transfer learning on the use of indoor localization based on thermal imaging. Neurocomputing 2016; 173:83–92.
6. Orr RJ, Abowd GD, The smart floor: A mechanism for natural user identification and tracking. In: CHI'00 extended abstracts on Human factors in computing systems, ACM, 2000, pp. 275–276.
7. Müller SM, Hein A, Multi-target data association in binary sensor networks for ambulant care support. Int J Advances Networks Services 2016; 9(1 & 2):20–29.
8. Song L, Wang Y, Multiple target counting and tracking using binary proximity sensors: Bounds, coloring, and filter. In: Proc. ACM Int. Symp Mobile Ad Hoc Networking Computing, ACM, 2014, pp. 397–406.
9. Crandall AS, Cook DJ, Tracking systems for multiple smart home residents. In: Human Behavior Recognition Technologies: Intelligent Applications for Monitoring and Security, Hershey, PA: IGI Global, 2011, pp. 111–129.
10. Hsu KC, Chiang YT, Lin GY, Lu CH, Hsu JYJ, Fu LC, Strategies for inference mechanism of conditional random fields for multiple-resident activity recognition in a smart home. In: Inter-

national Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Springer, 2010, pp. 417–426.
11. Wilson DH, Atkeson C, Simultaneous tracking and activity recognition (STAR) using many anonymous, binary sensors. In: Pervasive Computing, Berlin, Heidelberg: Springer, 2005, pp. 62–79.
12. Zafari F, Gkelias A, Leung KK, A survey of indoor localization systems and technologies. IEEE Communications Surveys & Tutorials 2019; 21(3):2568–2599.
13. Kim KW, Kwon J, Lee CG, Han J, Accurate indoor location tracking exploiting ultrasonic reflections. IEEE Sensors Journal 2016; 16(24):9075–9088.
14. Crandall AS, Cook DJ, Attributing events to individuals in multi-inhabitant environments. In: 2008 IET 4th International Conference on Intelligent Environments, 2008, pp. 1–8, doi:10.1049/cp:20081164.
15. Crandall AS, Cook DJ, Coping with multiple residents in a smart environment. Journal of Ambient Intelligence and Smart Environments 2009; 1(4):323–334.
16. Ghasemi V, Pouyan AA, Modeling users data traces in multi-resident ambient assisted living environments. Intl J Computational Intell Syst 2017; 10(1):1289–1297.
17. Amri MH, Becis Y, Aubry D, Ramdani N, Fränzle M, Robust indoor location tracking of multiple inhabitants using only binary sensors. In: 2015 IEEE Int. Conf. Automation Sci. Eng. (CASE), 2015, pp. 194–199, doi:10.1109/CoASE.2015.7294061.
18. De D, Song WZ, Xu M, Wang CL, Cook D, Huo X, FindingHuMo: Real-time tracking of motion trajectories from anonymous binary sensing in smart environments. In: 2012 IEEE 32nd Int. Conf. Distributed Computing Syst., 2012, pp. 163–172, doi:10.1109/ICDCS.2012.76.
19. Wang C, De D, Song WZ, Trajectory mining from anonymous binary motion sensors in smart environment. Knowledge-Based Syst 2013; 37:346–356.
20. Krishnan NC, Cook DJ, Activity recognition on streaming sensor data. Pervasive and Mobile Computing 2014; 10:138–154.
21. Kuhn HW, Variants of the hungarian method for assignment problems. Naval Research Logistics Quarterly 1956; 3(4):253–258.
22. Goodman IR, Mahler RP, Nguyen HT, Mathematics of data fusion. Springer Science & Business Media, 2013.
23. Gutmann M, Hyvarinen A, Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. Journal of Machine Learning Research 2012; 13:307–361.
24. Vo BN, Ma WK, The Gaussian mixture probability hypothesis density filter. IEEE Transactions on signal processing 2006; 54(11):4091–4104.