2011-01-01

# A Review of Situation Identification Techniques in Pervasive Computing

Juan Ye
*University of St. Andrews*

Simon Dobson
*University of St. Andrews*

Susan McKeever
*Dublin Institute of Technology*, susan.mckeever@dit.ie



## Recommended Citation

Ye. J, Dobson S., & McKeever S. (2012) A Review of Situation Identification Techniques in Pervasive Computing, *Pervasive and Mobile Computing* Vol. 8 no. 1, February, pp. 36-66 http://dx.doi.org/10.1016/j.pmcj.2011.01.004

# A Review of Situation Identification Techniques in Pervasive Computing

Juan Ye and Simon Dobson

*School of Computer Science, University of St Andrews, UK*
*email: ye@cs.st-andrews.ac.uk*

Susan McKeever

*School of Computer Science and Informatics, University College Dublin, IE*
*email: susan.mckeever@ucd.ie*

**Abstract**

Pervasive systems must offer an open, extensible, and evolving portfolio of services which integrate sensor data from a diverse range of sources. The core challenge is to provide appropriate and consistent adaptive behaviours for these services in the face of huge volumes of sensor data exhibiting varying degrees of precision, accuracy and dynamism. Situation identification is an enabling technology that resolves noisy sensor data and abstracts it into higher-level concepts that are interesting to applications. We provide a comprehensive analysis of the nature and characteristics of situations, discuss the complexities of situation identification, and review the techniques that are most popularly used in modelling and inferring situations from sensor data. We compare and contrast these techniques, and conclude by identifying some of the open research opportunities in the area.

*Keywords:* Pervasive computing, context modeling, situation identification, uncertain reasoning, temporal reasoning, machine learning, data mining, ontologies

## 1. Introduction

Pervasive computing embodies a vision of computers seamlessly integrating into everyday life, responding to information provided by sensors in the environment, with little or no direct instruction from users. It assumes a number of invisible sensing/computational entities that interact both with users and with the environment in which they operate. With the help of these entities, a pervasive computing system can deliver customised services to users in a context-aware manner when they are interacting and exchanging information with the environment. These days pervasive computing is maturing from its origin as an academic research topic to a commercial reality [55]. It has many potential applications, from intelligent workplaces and smart homes to healthcare, gaming, leisure systems and to public transportation [30]. These applications have significant potential to benefit human lives.

Sensors in pervasive computing are deployed anywhere and on any objects or human bodies. They collect data including a user's location, motion, biomedical information, environment temperature, humidity, or ambient noise level. Applications that provide customised services to users are based on this sensor data. However, sensor data exhibits high complexity (different modalities, huge volumes, and inter-dependency relationships between sources), dynamism (real-time update and critical ageing), accuracy, precision and timeliness. A pervasive computing system should therefore not concern itself with the individual pieces of sensor data (which room the user is in, what his heart rate or blood pressure is): rather, this information should be interpreted into a higher, domain-relevant concept, such as whether the user is suffering a heart attack or exercising. This higher-level concept is called a *situation*, which is an abstract state of affairs interesting to applications [31].

The power of using situations lies in their ability to provide a simple, human understandable representation of sensor data to applications, whilst shielding applications from the complexities of sensor readings, sensor data noise and inferences activities, and simultaneously leveraging the structure implicit in the activities being observed.

However, in anything beyond a small-scale system, there may be tens or hundreds of situations that applications need to recognise and respond to. Underlying these situations will be an even greater number of sensors that are used in situation identification. A system has a significant task of defining and managing these situations. This includes capturing *what* and *how* situations are to be recognised from which pieces of contexts, and how different situations are related to each other. The system should know, for example, which situations can or cannot occur at the same time (such as a user 'actively watching TV' and 'taking a shower' at the same time); otherwise, inappropriate adaptive behaviour may occur. Likewise, temporal order between situations may be important, such as the inability of a user to go directly from a situation of 'sleeping' to 'going for a run'. Given the inherent inaccuracy of sensor data and the limitations of inference rules, the detection of situations is imperfect.

This paper provides a comprehensive understanding of situations in pervasive computing. Section 2 introduces the definitions of situations, their features, the general research topics on situations, and the factors that make research on situations challenging. Section 3 reviews the main stream of works around situation identification, including formal logic methods, ontology-based situation models, machine learning, and data mining techniques, which are elaborated in Section 4 and Section 5 respectively. Section 6 provides a qualitative evaluation on the reviewed works in terms of identifying situations at different levels of abstraction; the ability to resolve uncertainty; the ability to represent and reason on temporal information; the ability to deal with complex situations; the ability to incorporate and derive knowledge; the knowledge engineering effort involved to be able to use these techniques; and the effect of different sensing technologies on choosing situation identification techniques. Based on this analysis, we discuss the future opportunities in situation identification research in Section 7. Section 8 concludes the paper.

## 2. Overview of Situation identification in Pervasive Computing

For clarity we shall define some of the terms that will appear frequently later. *Sensor data* encompasses raw (or minimally-processed) data retrieved from both physical sensors and 'virtual' sensors observing digital information such as user calendars and network traffic. This data is aggregated to form *context* – the environment in which the system operates, understood symbolically – which may be further sub-divided into context derived directly from sensors (*primary* context) and that inferred and/or derived from several data streams (*secondary* context). An important form of secondary context is *activities* representing small, higher-level inferences about contextual information, such as the activity of 'cutting' derived by observing motion over time [102]. Finally, a *situation* is an abstraction of the events occurring in the real world derived from context and hypotheses about how observed context relates to factors of interest to designers and applications. Situations typically fuse several sources of context, as well as domain knowledge, spatial and temporal models of the expected behaviour of the phenomena being observed.

### *2.1. Sensors and Sensor Data*

Service provision of a pervasive computing system relies on the perception of an environment, supporteded by a range of sensors. Sensing technologies have made significant progress on designing sensors with smaller size, lighter weight, lower cost, and longer battery life. Sensors can thus be embedded in an environment and integrated into everyday objects and onto human bodies. Sensors in pervasive computing can capture a broad range of information on the following aspects [30]:

- *Environment*: temperature, humidity, barometric pressure, light, and noise level in an ambient environment and usage of electricity, water, and gas;
- *Device*: state of devices (such as available or busy), functions of devices (such as printing or photocopying), the size of memory, the resolution of screen, or even embedded operating systems;
- *User*: location, schedule, motion data like acceleration of different parts of bodies, and biometrical data like heart rate and blood pressure;

- *Interaction*: interacting with real objects through RFID and object motion sensors [77], and interacting with devices through virtual sensors like monitoring frequencies of a user using his keyboard and mouse [91, 98].

The diversity of sensors leads to high complexity in interpreting their output, including huge data volumes, different modalities, inter-dependence, real-time update, and critical ageing. In dealing with the real world, these sensors typically produce imperfect data. Noisy sensor data may result in misunderstanding of a user's or an environmental state, which will lead to incorrect application behaviour. These sensors also have their own technical limitations, are prone to breakdown, may be disconnected from the sensor network, or be vulnerable to environmental interference. This leads to the uncertainty issue of sensor data, which can be *out of date*, *incomplete*, *imprecise*, and *contradictory* with each other [54]. These features of pervasive sensor data complicate the process of making themselves immediately understandable or usable to applications. A pressing challenge is therefore how to use them in recognising patterns that could give us a better understanding of human interactions with an environment [4].

Different sensors produce different types of sensor data, including *binary*, *continuous numeric*, and *featured* values. The types of data will have an impact on techniques chosen to analyse them. A binary value is the simplest type of sensor data: *true* (1) or *false* (0). RFID sensors produce a binary reading: an object with a RFID tag is detected by a reader or not; or a binary-state sensor developed in University of Amsterdam [137] produces 1 when it is fired. Continuous numeric values are produced by most sensor types, including positioning sensors, accelerometers, and all the ambient sensors. Featured values are typically produced from relatively more sophisticated sensors such as a camera and a eye movement tracker, whose data needs to be characterised into a set of categorical measurements. For example, motion features can be extracted from video streams recorded in cameras, including quantity of motion and contraction index of the body, velocity, acceleration and fluidity [21]. Eye movements captured in Electrooculography signals are characterised into two types of features: *saccades* that are the simultaneous movement of both eyes in the same direction and *fixations* that are the static states of the eyes during which gaze is held upon a specific location [20]. Table 1 summarises the commonly used sensors and their types of sensor data.

| **Sensor types** | **Type of values** |
|---|---|
| Interaction with objects | binary, numeric |
| Location detection | numeric |
| Acceleration | numeric |
| Eye movement | features |
| Biometric parameters | numeric |
| Resource usage | numeric |
| Ambient parameters | numeric, binary |
| Processed video | features |

Table 1: A list of commonly used sensors in smart environments.

### *2.2. Situations*

A general architecture of information flow in a pervasive computing system is described in figure 1, from which we derive situations and relate situations with other types of information in the system. We use a typical scenario in pervasive computing as an example – a healthcare monitoring system in a smart home.

At the bottom of figure 1, sensors produce data, which can be abstracted into a set of domain concepts, typically called *context*. To distinguish context from raw sensor data, we regard context as a well-structured concept that describes a property of an environment or a user. Contexts can be classified into different domains in terms of the property they describe. Contexts in one domain can also have a different structure, which is distinguishable from contexts in other domains. For example in the domain of location a coordinate context can be identified as three numeric values measured in meters, while in the domain of temperature a context can be identified as a numeric value with its unit of measurement as Celsius or Fahrenheit. A

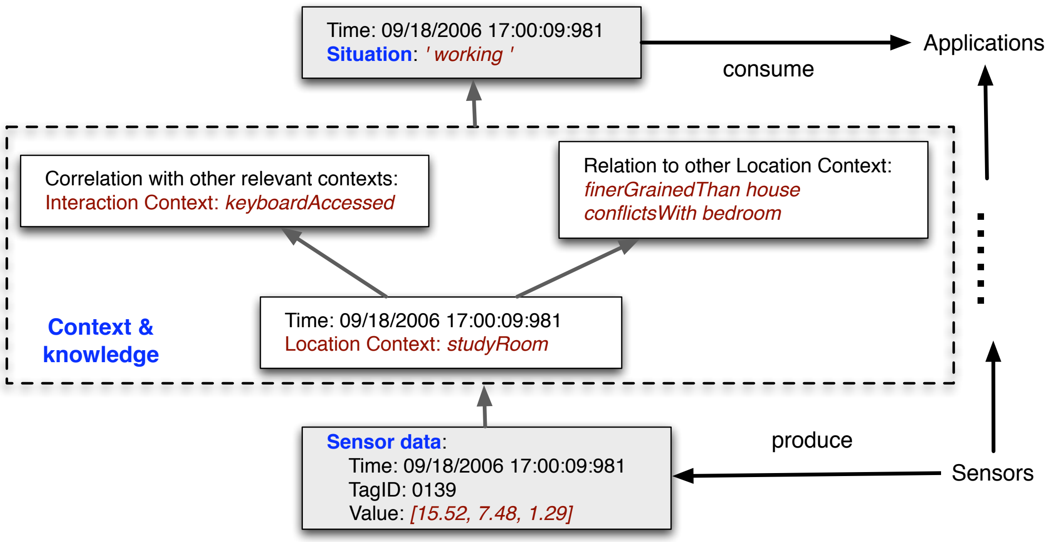


Figure 1: Information flow in pervasive computing

secondary context can be considered as a characteristic function on sensor data: for example in figure 1, a symbolic location context `studyRoom` is mapped to a set of coordinate contexts that are valid inside a study room.

Semantic relationships exist between contexts in one domain, including *different levels of granularity*, *overlapping*, and *conflicting* [120, 152]. The semantic relationships can be explored by applying *domain knowledge*, or through a learning process, or both. Here, domain knowledge is regarded as knowledge specific to available context, including spatial knowledge for location contexts, or user preferences for person contexts. For example, given a map of a house, we can specify the spatial relationships between the location contexts as: the location context `studyRoom` is finer grained than another location context `house` (*spatially contained in*), and it conflicts with the context `bedroom` (*spatially disjoint to*).

A *situation* is defined as an *external semantic interpretation* of sensor data. *Interpretation* means that situations assign meanings to sensor data. *External* means that the interpretation is from the perspective of applications, rather than from sensors. *Semantic* means that the interpretation assigns meaning on sensor data based on structures and relationships within the same type of sensor data and between different types of sensor data.

A situation can be defined by collecting relevant contexts, uncovering meaningful correlations between them, and labelling them with a descriptive name. The descriptive name can be called a *descriptive definition* of a situation, which is about how a human being defines a state of affairs in reality. A logical expression of correlated context predicates is called a *logical specification* of a situation. With these two, a situation bridges sensor data and applications. Sensor data is abstracted to a certain situation by evaluating its specification, and this situation will trigger applications that correspond to its descriptive name.

For example in figure 1, the location context `studyRoom` conjuncts with an interaction context `keyboardAccessed` indicating that the keyboard of the desktop located in the study room is accessed. These two contexts are considered relevant in that they share the same location semantics – the study room. Their correlation forms a logical specification of a situation with its descriptive definition as 'working', meaning that if a user is in the study room and accessing the keyboard of the desktop, he or she is considered in a 'working' situation. An application can be defined on this situation; for example, adjusting the sound level of the background music. At a certain time, if the sensor data input from positioning and interaction sensors satisfy the conditions of this situation, the application behaviour associated with this situation will be executed automatically.

*2.2.1. Features of Situations*

A situation is a subjective concept, whose definition depends on *sensors* in a current system, which decide available contexts used in a specification; on *the environment* where the system works, which determines the domain knowledge to be applied (e.g., a spatial map); and on the requirement of *applications*, which determines what states of affairs are interesting.

The same sensor data can be interpreted to different situations according to requirements of applications. For example, based on the location data for a number of users, we can define (1) *user-centered* situations (`meeting` – the users are gathering in a meeting room), and (2) *location-centered* situations (`occupied` – a room is occupied). A situation is a particular state that is abstracted from sensor data and is interesting to applications so that certain actions can be taken when this situation is occurring.

What distinguishes situations from activity, and situation recognition from activity recognition, is the inclusion in situations of rich temporal and other structural aspects, including *time-of-day* – a situation may only happen at a particular time of the day; *duration* – it may only last a certain length of time; *frequency* – it may only happen a certain times per week, and *sequence* – different situations may occur in a certain sequence. A situation can be a simple, abstract state of a certain entity (e.g., a room is occupied), or a human action taking place in an environment (e.g., working or cooking). A situation can also be composed of or abstracted from other finer-grained situations; for example, a 'seminar' situation includes the finer situations like 'presentation', 'questioning', and 'group discussion'.

Rich relationships exist between situations, including:

**Generalisation** A situation can be regarded more *general* than another situation, if the occurrence of the latter implies that of the former; for example, a 'watching TV' situation is considered more specific than an 'entertainment' situation, because the conditions inherent in the former situation subsume or imply the conditions in the latter situation [149].

**Composition** A situation can be decomposed into a set of smaller situations, which is a typical *composition* relation between situations. For example, a 'cooking' situation is composed of a 'using stove' situation and a 'retrieving ingredients' situation. McCowan *et al* propose a two-layered framework of situations: a group situation (e.g., 'discussion' or 'presentation') is defined as a composition of situations of individual user (e.g., 'writing' or 'speaking') [89].

**Dependence** A situation *depends* on another situation if the occurrence of the former situation is determined by the occurrence of the latter situation. Dependence can be long- or short-range, as proposed by Choujaa and Dulay [26]. Sometimes long-range dependence can be more useful in inferring high-level situations. For example, a situation 'going to work' may be better in inferring a situation 'going home from work' than other short-range dependent situations.

**Contradiction** Two situations can be regarded as *mutually exclusive* from each other if they cannot co-occur at the same time in the same place on the same subject; for example, a user cannot be in a cooking situation and a sleeping situation at the same time.

**Temporal Sequence** A situation may occur before, or after another situation, or interleave with another situation; for example, 'taking pill' should be performed after 'having dinner' [62].

*2.3. Research Topics on Situation Identification*

In pervasive computing, the principal research topics on situation identification involve the following issues:

- **Representation** how to define logic primitives that are used to construct a situation's logical specification.

- **Specification** how to form a situation's logical specification, which can be acquired by experts or learned from training data;

- **Reasoning** how to infer situations from a large amount of imperfect sensor data; how to reason on situations' relationships; and how to maintain the consistency and integrity of knowledge on situations.

Unlike the well-known situations used in the Natural Language Processing domain, situations in pervasive computing are highly related to sensor data, domain knowledge on environments and individual users, and applications. As discussed in the above sections, sensor data occur in large volumes, in different modalities, and are highly inter-dependent, dynamic and uncertain. Situations are in a rich structural and temporal relationship, and they evolve in diffuse boundaries. In addition, the complexity in domain knowledge and applications makes studying situations a very challenging task.

In *representation*, logical primitives should be rich enough to capture features in complicated sensor data (e.g., acceleration data), domain knowledge (e.g., a spatial map or social network), and different relationships between situations. Also a pervasive computing system is assumed to be highly dynamic in the sense that it might introduce new sensors that yield new types of context, so the logical primitives should be flexibly extensive; that is, new primitives will not cause modifications or produce ambiguous meanings on existing ones [148].

In *specification*, it is difficult for experts to locate relevant contexts to a situation, decide their different contribution weights (i.e., to what degree the contexts contributes to identifying a situation), and quantify their uncertainty measurements (i.e., to what degree the input sensor data validate the contexts).

In *reasoning*, one of the main processes is called *situation identification* – deriving a situation by interpreting or fusing several pieces of context in some way. The performance of reasoning is usually undermined by the complexity of the underlying sensor data.

The diversity of applications complicates these issues even more. One of the main requirements of a pervasive computing system is to deliver correct services to correct users at the correct places at the correct time in a correct way. It is assumed that a system should host a large number of applications that can be finely tuned for different situations. This requires a situation model to support evolution of situations' specifications and to be able to maintain consistency between original and evolving specifications.

These applications can also have different degrees of significance to the system, user, or environment. Some applications can only be triggered if a situation is critical and the confidence of identifying this situation is high; for example in a smart home environment, an application could be to make the emergence call when the house is in a fire or electrical accident or the occupant suffers heart attack. This type of applications will be triggered if these hazardous situations are inferred, even if inferred with a lower confidence relative to other situations. The situation model must not only be able to handle uncertainty, but also be informative about inference results; that is, what situations are most likely to happen while what situations are possible or impossible to happen [80].

This section has introduced the basic elements of information flow in pervasive computing: sensors, contexts, situations, and applications. It has described the research on situation identification and the impact of the characteristics of sensors and applications on this research. In the following, we will provide an overview of the existing techniques that have been popularly applied in the research of situation identification.

## 3. Situation Identification Techniques

Situation identification techniques have been studied extensively in pervasive computing, and here we highlight those techniques we consider to show the most promise. Figure 2 shows the development of the situation identification techniques and their correlation to the increasing complexity of problem descriptions.

Figure 2: Development of the main situation identification techniques corresponding to the increasing complexity of problem descriptions

### *3.1. Specification-based Approaches*

In the early stages, situation identification research starts when there are a few sensors whose data are easy to interpret and the relationships between sensor data and situations are easy to establish. The research consists mainly of specification-based approaches that represent expert knowledge in logic rules and apply reasoning engines to infer proper situations from current sensor input. These approaches have developed from earlier attempts in first-order logic [46, 111] towards a more formal logic model [80] that aims to support efficient reasoning while keeping expressive power, to support formal analysis, and to maintain the soundness and completeness of a logical system. With their powerful representation and reasoning capabilities, ontologies have been widely applied [23, 113, 127, 148]. Ontology-based approaches can be considered complementary to formal logic approaches in that ontology can provide a standard vocabulary of concepts to represent domain knowledge, specifications and semantic relationships of situations defined in formal logic approaches and provide full fledged reasoning engines to reason on them following axioms and constraints specified in formal logic approaches.

As more and more sensors are deployed in real-world environments for a long term experiment, the uncertainty of sensor data starts gaining attention. To deal with the uncertainty, traditional logic-based techniques need to be incorporated with other probabilistic techniques [32]:

$$certainty = \Sigma_{i=1}^{n} w_i \mu(x_i) \tag{1}$$

where *certainty* is the certainty associated with an inferred situation, $n$ is the number of conditions that contribute to identification of this situation, $w_i$ is the weight on a certain condition, and $\mu(x_i)$ is the degree that the condition is satisfied by the current sensor data.

The above general formula uncovers two issues in situation identification. First, the satisfaction of a condition is not crisply either true or false, which should take into account the imprecision of sensor data. Fuzzy logic, with its strength in dealing with imprecision, has been applied to solving this issue [3]. Secondly, not every condition contributes to identifying a situation to the same degree, so the problem becomes how to identify the significance of each evidence, how to resolve conflicting evidences, and how to aggregate evidences. Evidence theories like Dempster-Shafer Theory have been used to solve this problem [58, 90].

### *3.2. Learning-based Approaches*

Moving towards the right hand side of Figure 2, advances in sensor technologies boosts the deployment of a broad range of sensors, which however undermine the performance of specification-based approaches. It is less feasible to only use expert knowledge to define proper specifications of situations from a large number of noisy sensor data. To address this problem, techniques in machine learning and data mining are borrowed to explore association relations between sensor data and situations. A large amount of the research has been conducted in the area of activity recognition in smart environments recently.

A series of Bayesian derivative models are popularly applied, including Naïve Bayes [106, 129] and Bayesian networks [44, 111] with the strength in encoding causal (dependence) relationships, and Dynamic Bayesian Networks [136], Hidden Markov Models [53, 143] and Conditional Random Fields [134, 137] with the strength in encoding temporal relationships. Inspired from the language modelling, grammar-based approaches like (stochastic) context free grammars are applied to representing the complex structural semantics of processes in hierarchical situations [96, 117, 132]. Decision trees [10, 77], Neural Networks [145], and Support Vector Machines [65, 104] as another branch in machine learning techniques, which are built on information entropy, have also been used to classify sensor data into situations based on features extracted from sensor data.

Even though the above learning techniques have achieved good results in situation identification, they need a large amount of training data to set up a model and estimate their model parameters [130]. When training data is precious, researchers are motivated to apply web mining techniques to uncover the commonsense knowledge between situations and objects by mining the online documents; that is, what objects are used in a certain human activity and how significant the object is to identifying this activity [103, 109, 128]. Some unsupervised data mining techniques have been applied as well, including Suffix-tree [49, 50] and Jeffrey divergence [17, 18].

## 4. Specification-based Techniques

Specification-based approaches typically build a situation model with *a priori* expert knowledge and then reason on it with input sensor data. This section will introduce the main stream of specification-based techniques.

### *4.1. Logic Programming*

The key underlying assumption of formal logic approaches is that knowledge about situations can be modularised or discertised [80]. The goal of these approaches is to provide a theoretical foundation for building a situation-aware system with the capabilities of (1) formally representing logic specifications of situations; (2) verifying the integrity and consistency of situation specifications in a rule base; and (3) extending existing systems to be able to incorporate more sensor data and recognise more situations.

This strand of work starts from using predicate logic in defining situations, including Ranganathan et al [111], Henricksen et al [55], Gu et al [45, 46], to name a few. These works take the initiative in considering situations as the abstraction of sensor data that would be more influential on applications. They have worked on how to define situations in simple logical formulas using purely expert knowledge. Take an example from Henricksen and Indulska [55], which defines an `Occupied` situation when a person is detected to be engaged in an event that generally should not be interrupted; for example, 'in meeting' or 'taking call'. This situation is specified in the following code:

$$
\begin{aligned}
Occupied(person) = \exists t_1, t_2, event | engagedin(person, event, t_1, t_2) \cdot \\
((t_1 \leq timenow() \wedge (timenow() \leq t_2 \vee isnull(t_2))) \vee \\
((t_1 \leq timenow() \vee isnull(t_1)) \wedge timenow() \leq t_2))) \wedge \\
(event = "inmeeting" \vee event = "takingcall")
\end{aligned}
$$

These initial works have been advanced to a more formal approach by Loke et al [15, 78, 79, 80]. They have proposed a declarative approach to represent and reason with situations at a high level of abstraction. This approach is based on the logical programming language that embeds situation programs in Prolog, which provides developers a high level of programming and reasoning on situations. For example [79],

```
if in_meeting_now(E) then
   with_someone_now(E) ,
   has_entry_for_meeting_in_diary(E) .
if with_someone_now(E) then
   location*(E, L) , people_in_room*(L, N) , N > 1.
if has_entry_for_meeting_in_diary(E) then
   current_time*(T1) ,
   diary*(E, 'meeting', entry(StartTime, Duration)) ,
   within_interval(T1, StartTime, Duration) .
```

In the above situation program, a situation `in_meeting_now` of a user entity `E` is defined on two situations `with_someone_now` and `has_entry_for_meeting_in_diary`. Each of these situations has its own program that is defined on sensor predicates; for example, `with_someone_now` refers to two sensor predicates: `location*(E, L)` that returns the location of the entity, and `people_in_room*(L, N)` that returns the number of people in the location. In this way, situation programs are made amenable to formal analysis, and the inference procedure of reasoning about situations is decoupled from the acquisition procedure of sensor readings. This modularity and separation of concerns facilitates the development of context-aware systems [79]. Based on situation programs, developers can perform meta-level reasoning over them, including determining the most likely current situation, specifying relations on situations (such as composition and concurrency), deciding appropriate situation-driven behaviours, and evaluating soundness and completeness of situation programs [78].

Mature logic theories in other disciplines have also been used. For example, *situation calculus* (also called situation theory) [87, 88] has long been a staple of AI research. It is a logic language to AI, where a situation is considered as a snapshot of the real world at some instant. It is designed to specify and implement a dynamical system in that it has a capability in representing and reasoning on actions that lead to situation changes [73]. With its solid foundation in logic and power in expressing rich semantics, it has been borrowed to study situations in pervasive computing [64, 72, 94].

*4.2. Spatial and Temporal Logic*

As pointed out by Cook *et al*, very little can be done without an explicit or implicit reference to where and when the meaningful situations occurred [29]. Spatial and Temporal logic is a well established area of AI, which has been applied to representing and reasoning on spatial and temporal features and constraints of context and situations.

Augusto *et al* introduce the temporal operators `ANDlater` and `ANDsim` in Event-Condition-Action rules, upon which temporal knowledge on human activities can be specified [6]. The following rule specifies a situation of a user 'fainting', where the sensor event `at_kitchen_on` represents the activation of the RFID sensors in the kitchen, `tkRK_on` represents the activation of the RFID sensor while the user is passing through the door between the kitchen and the reception area, and `no_movement_detected` represents no detection of any movement [6].

```
IF at_kitchen_on ANDlater tdRK_on ANDlater no_movement_detected
THEN assume the occupant has fainted
```

| *Relation* | *Illustration* | *Interpretation* |
|---|---|---|
| $I_1 < I_2$<br>$I_2 > I_1$ | $I_1$ $I_2$ | $I_1$ before $I_2$<br>$I_2$ after $I_1$ |
| $I_1 \mathsf{m} I_2$<br>$I_2 \mathsf{mi} I_1$ | $I_1$ $I_2$ | $I_1$ meets $I_2$<br>$I_2$ met by $I_1$ |
| $I_1 \mathsf{o} I_2$<br>$I_2 \mathsf{oi} I_1$ | $I_1$ $I_2$ | $I_1$ overlaps $I_2$<br>$I_2$ overlapped by $I_1$ |
| $I_1 \mathsf{s} I_2$<br>$I_2 \mathsf{si} I_1$ | $I_1$ $I_2$ | $I_1$ starts $I_2$<br>$I_2$ started by $I_1$ |
| $I_1 \mathsf{d} I_2$<br>$I_2 \mathsf{di} I_1$ | $I_1$ $I_2$ | $I_1$ during $I_2$<br>$I_2$ contains $I_1$ |
| $I_1 \mathsf{f} I_2$<br>$I_2 \mathsf{fi} I_1$ | $I_1$ $I_2$ | $I_1$ finishes $I_2$<br>$I_2$ finished by $I_1$ |
| $I_1 = I_2$ | $I_1$ $I_2$ | $I_1$ equals $I_2$ |

Figure 3: Thirteen temporal relations [41]

Gottfried *et al* apply qualitative AI techniques in dealing with temporal and spatial knowledge in smart homes [41]. Allen's Temporal Logic is utilised to describe, constrain, and reason on temporal sequences between two events, as shown in Figure 3. For example, given the following events: the door to the room is opened ($I_1$) before the person is in the room ($I_2$) and the sensor is triggered ($I_3$) during the person's presence in the room, then the composition of these events suggests that the door is opened before the sensor is triggered, which is inferred as follows: $I_1 < I_2$ and $I_2$ $di$ $I_3$ then $I_1 < I_3$, where $di$ indicates the temporal relationship 'duration'. Region Connection Calculus is used to describe spatial relationships between regions and spatial arrangement of objects in the regions. For example, 'externally connected to'– `EC(living room, hallway)`, and 'tangential proper part of' – `TPP(sofa, living room)`. Distance properties can be combined with these topological relations between regions and movement trajectory primitives, which are used to describe primitive activity patterns so that more sophisticated activity patterns can be distinguished. For example in Figure 4, the top box represents a characterised movement trajectory. After combining with region and distance primitives, this trajectory can imply two different activity patterns: a hectic running to and fro in the passage (as shown in Figure 4 (a)) and a systematic searching in the lounge (as shown in Figure 4 (b)).

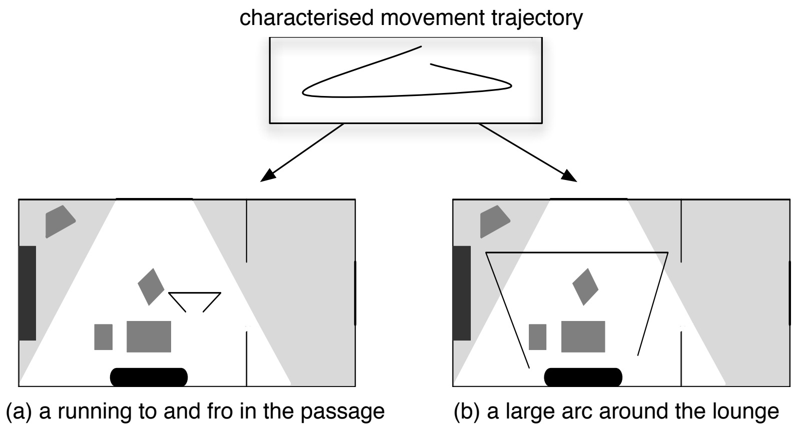


Figure 4: An example of inferring activities from the combination of a movement trajectory and the information about distances [41]

*4.3. Ontologies*

Ontologies have increasingly gained attention as a generic, formal and explicit way to 'capture and specify the domain knowledge with its intrinsic semantics through consensual terminology and formal axioms and constraints' [148]. They provide a formal way to represent sensor data, context, and situations into well-structured terminology, which makes them understandable, sharable, and reusable by both humans and machines [24].

Ontologies support a set of modelling primitives to define classes, individuals, attribute properties, and object properties (i.e., relations between objects). For example, the `is-a` property is one of the most useful properties in modelling the abstraction level of two domain concepts: `'Dining Room' is-a 'Eating Activity Space'` [144], and `'Pan' is-a 'Cooking Utensil'` [128]. Ontologies are expressive in modelling entities in a smart environment and domain concepts, including sensors [125], complex sensor events (like video events by Luo and Fan [83], Nevatia et al. [100], SanMiguel et al. [118]), a space map of the environment [126, 144], user profile [70], objects that users interact with [128], ambient information such as temperature, humidity, and sound level [46].

Based on the modelled concepts, developers can define logical specifications of situations in rules. An exemplar rule on the activity 'sleeping' is taken from the work by [45]:

```
(?user rdf:type socam:Person),
(?user, socam:locatedIn, socam:Bedroom),
(?user, socam:hasPosture, 'LIEDOWN'),
```

```
(socam:Bedroom, socam:lightLevel, 'LOW'),
(socam:Bedroom, socam:doorStatus, 'CLOSED')
-> (?user socam:status 'SLEEPING')
```

Ontologies with the representation formalisms can be used to support reasoning, including (1) detecting inconsistency; (2) deriving new knowledge [13]. When developers have built ontologies, an ontological reasoner can be used to check consistency in a class hierarchy and consistency between instances; that is, whether a class being a subclass of two classes that are declared as disjoint; or whether two instances are contradictory to each other; for example, a person being in two spatially disjoint locations at the same time. Given the current sensor data, the reasoner will derive a new set of statements. In the above 'sleeping' example, if the reasoner is based on a forward-chaining rule engine, it can match the conditions of this rule against the sensor input. If all the conditions are satisfied, the reasoner will infer the conclusion of the rule. The reasoning will terminate if the status of the user is inferred, when the status of the user is set to be the default inference goal in this reasoner.

Another example of using ontologies in activity recognition is given by [115], as shown in Figure 5. Instead of using ontologies to infer activities, they use the ontologies to validate the result inferred from statistical techniques. In Figure 5, the top box provides the pre-defined knowledge; that is, the relationships between the domain concepts including rooms, objects, and activities. For example, the symbolic location 'RestRoom' is defined as `is-a` 'Room' with a 'Sink', and the other symbolic location 'LivingRoom' `is-a` 'Room' without a 'WaterFixture', while a 'Sink' `is-a` a 'WaterFixture'. An activity 'BrushTeeth' is defined as `is-a` 'PersonalActivity' and must be performed in a location with a 'Sink'. Based on this pre-defined knowledge, the ontological reasoner can derive a set of locations where the activity 'BrushTeeth' can be performed, as shown in the bottom box. Assume that the sensors report the user's current location as 'LivingRoom' and a statistical technique infers the current possible activities as $\{$(BrushTeeth 0.6), (Reading, 0.5)$\}$. Filtered with the new derived knowledge, the system will derive the activity 'Reading' as the inference result, even though 'BrushTeeth' has the highest probability.

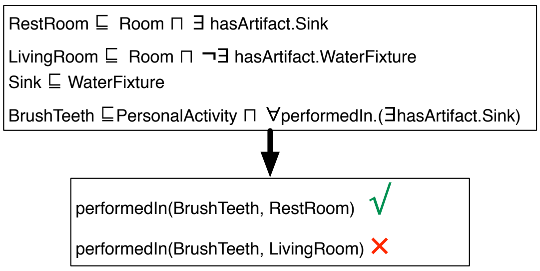


Figure 5: An example of the use of ontologies in deriving new facts

Ontologies have also been used in modelling complex concepts and reasoning on their relationships such as Video events [83, 100, 118] and complicated activities [12, 23, 31, 46, 84, 97, 124, 140, 146]. Nevatia *et al* introduce a formal language for describing an ontology of events, called the Video Event Representation Language, *VERL* [100]. This representation of video events aims to support tasks like video surveillance, video browsing, and content-based video indexing. Yau *et al* [146] develop a situation ontology, which incorporates rich semantics (including *time constraint*) in writing logical specifications of situations.

Beyond the semantic enrichment at a conceptual level, ontologies can also serve as an effective basis for the development of knowledge-centric software; for example, agent frameworks and middleware for context-aware systems [24, 67, 70]. Chen et al. [24] propose a layered conceptual architecture for semantic smart homes. The novelty of this architecture is that there exists a Semantic layer and an Intelligent Service layer between the Data and Application layers. The Semantic layer achieves data interoperability and machine understandability by using ontologies to provide a homogeneous view over heterogeneous data, while the Intelligent Service layer delivers the capability of interoperability and high level automation by exploiting semantics and descriptive knowledge in the ontologies.

All the above logical approaches including ontologies are a formal approach of representing semantics of context and situations – the knowledge box in Figure 1. A considerable amount of knowledge engineering effort is expected in constructing the knowledge base, while the inference is well supported by mature algorithms and rule engines. For example, Gu *et al* use a forward-chaining rule engine in performing reasoning, which is based on a standard RETE algorithm [45].

### *4.4. Fuzzy Logic*

The theory of fuzzy sets is widely used to deal with uncertainty of vagueness, which is represented in a membership function – to what degree an element belongs to a fuzzy set [154]. There are three kinds of interpretations of (fuzzy) degree of a membership function. It could be the degree of similarity that represents the degree of proximity of the element to prototype elements of the fuzzy set. The elements are grouped by their relative proximity to each prototype. It could also be the degree of preference that represents an intensity of preference in favour of one over some other possible choices. The third type of interpretation is the degree of possibility that a parameter (variable) has a certain value.

In pervasive computing, fuzzy logics are used to map sensor data to linguistic variables that make sense to human social or conceptual settings [32] and to evaluate the similarity degree of concepts [3]. A probability value reflects the reliability of data, which can be provided from the domain expert, or obtained from the user experience. It allows imprecise knowledge to be modelled, such that an approximation of a numerical value, or a vague information is expressed. For example in the domain of Temperature, fuzzy functions on temperature terms like 'cold', 'lukewarm', and 'warm' can be mapped to a (possibly overlapping) range of temperature degrees. A temperature degree $15°C$ could be evaluated against these fuzzy functions and inferred as 'cold' with the fuzzy value 0.3, 'lukewarm' with 1.0, and 'warm' with 0.1. It is the preferred way to deal with knowledge numerically provided by measuring devices, and knowledge symbolically expressed by a human observer [56]. Fuzzy logic supports the operations including intersection, union, complement and modifier of fuzzy sets.

Delir *et al* have introduced fuzzy functions to characterise sensor data; to what degree the input sensor data (e.g., the systolic or diastolic blood pressure) matches to the linguistic variables (e.g., low, normal, or high) that are used in situation definitions (e.g., hypotension, normal, or hypertension) [32]. Anagnostopoulos *et al* [3] apply fuzzy inference on a conceptual hierarchy of situational context ontologies, as shown in Figure 6. These ontologies describe and interpret the specific contextual information (such as Temporal, Spatial, and Artefact context) associated with the user situation (such as 'Meeting', 'Formal Meeting', and 'Jogging'). The inference process is to find a situation that is most similar to the current unknown situation by evaluating the similarity of the specifications of situations in the knowledge base and the current context input. A fuzzy function is applied to evaluate the degree of membership in a *situational involvement* that refers to the degree of belief that a user is involved in a recognised/estimated situation [152].. This situational involvement measurement will be used to help a system to decide whether or which tasks should be executed to react to the inferred situation.

### *4.5. Evidence Theory*

Dempster-Shafer Theory (DST) [33, 121, 122] is a mathematical theory of evidence, which propagates uncertainty values and consequently provides an indication of the certainty of inferences. The core concepts of DST are the mass functions, the frame of discernment and combination rules. Mass functions distribute belief from sensors across choices or hypotheses in the frame of discernment. The combination rule is used to to fuse belief from multiple sources.

As a generalised probability approach, DST has distinct features: it quantifies and preserves ignorance due to the lack of information and it aggregates beliefs when new evidence is accumulated. One of its important aspects is the combination of evidence obtained from multiple sources and the modelling of conflict between them. As for the model itself, the significant innovation of DST is that it allows for the allocation of a probability mass to sets or intervals. Unlike probabilistic techniques, it does not force belief to be allocated to singletons, but only allocated belief according to the knowledge available. The model is designed to cope with varying levels of precision regarding the information and no further assumptions are

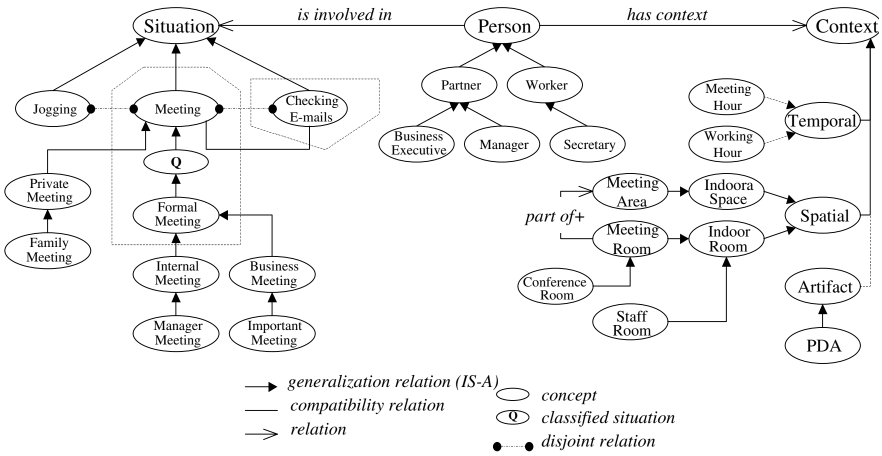


Figure 6: Situational context ontology [3]

needed to represent the information. This is potentially valuable for the evaluation of risk and reliability in engineering applications when it is not possible to obtain a precise measurement from experiments, or when knowledge is obtained from expert elicitation.

With its strengths in quantifying uncertainty and distributing belief, DST has been applied in activity recognition, such as [58, 90, 93, 121, 155]. The basic premise of using DST for situation identification is as follows: sensor readings are used as evidence of higher level states or context within an activity model; and these states are fused to determine more complex and higher level states until the level of belief in the activities of interest is determined [93]. We will take the work of [58] as an example to illustrate the use of DST in activity recognition.

The process of using DST is described as follows. First of all, developers need to apply expert knowledge to construct an evidential network that describe how sensors lead to activities. The left-hand side of Figure 7 describes that the sensors on the cup and fridge (e.g., *scup* and $sfri$) are connected to context information (e.g., 'cup' and 'fridge'), while the context information can be further inferred or composed to higher-level context (e.g., *juice* and $cup, juice$). The composition of context information points to an activity (e.g., 'Making cold drink') at the top.

Secondly, developers need to determine the evidence space and degree of belief for each evidence. As shown in the right-hand side of Figure 7, from the bottom the mass function represent the belief distribution on characterised sensor observations (e.g., the possible sensor values on 'sfri' are $sfri$ and $\neg sfri$. The uncertainty of sensor observations is quantified in the discounted mass functions. Then the degrees of belief on sensor observations will be translated into the degrees of belief on the associated object context node (e.g., 'fridge') by using the multi-valued mapping: $\{sfri \rightarrow fridge\}$, $\{\neg sfri \rightarrow \neg fridge\}$, and $\{sfri, \neg sfri \rightarrow fridge, \neg fridge\}$. The context can be lifted to higher-level context (e.g, $fridge \rightarrow juice$), while the mass function on the original context will be propagated to the higher-level context through an evidential mapping: $\{fridge\} \rightarrow \{(\{juice\}, 0.9), (\{juice, \neg juice\}, 0.1)\}$; that is, when the sensor on the fridge fires, the probability that the user takes the juice is 90%. An *equally weighted sum operator* is applied to sum the belief distributions on component context nodes when they lead to a composite node ($\{cup\}$ and $\{juice\}$ to $\{cup, juice\}$). Dempster's combination formula is applied when aggregating context nodes to an activity node. In Figure 7, the activity node 'Making cold drink' is resulted from a single composite context node, so the mass function on the activity is the belief function on the context node.

McKeever *et al* extend this work with a more comprehensive modelling of sensor quality, the incorporation of temporal features of activities, and multiple evidence fusion operators. They take into accounts sensor data quality such as dynamic time decays and fuzzy imprecise state [92]. They use the duration of an activity

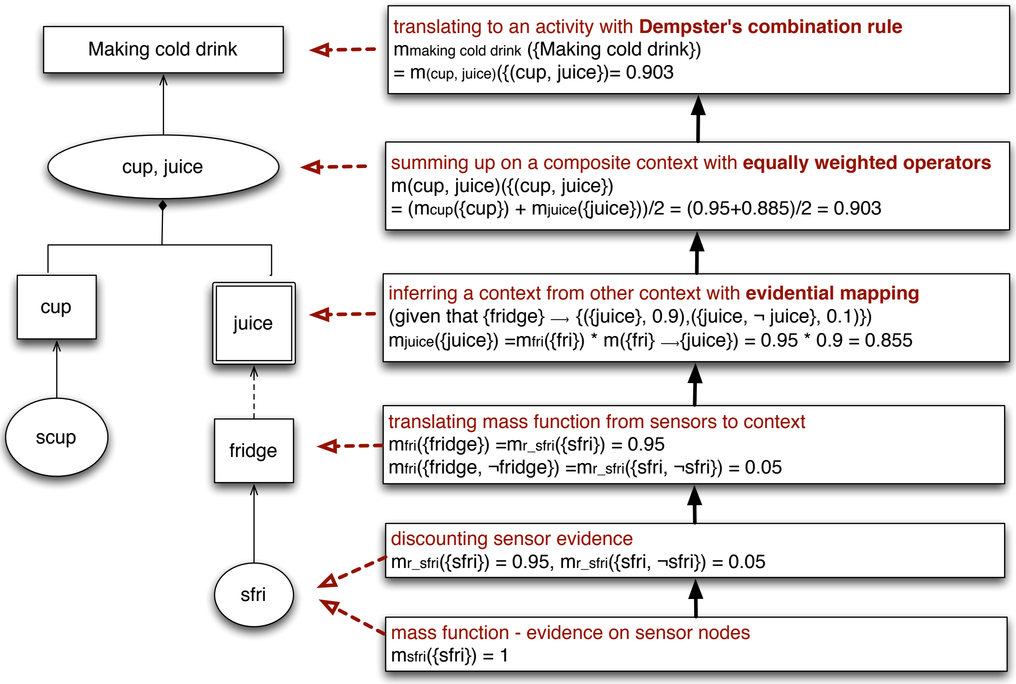


Figure 7: The use of Dempster-Shafer Theory

as the basis to extend the lifetime of lower-level evidence (e.g., sensor, context, or lower-level activity) and then fuse them in a *time extended* mass function [90].

The advantages of using DST in activity recognition are (1) it is able to represent multiple types of uncertainty; (2) it is human-understandable and explainable for how a system recognises activities. However, this technique heavily relies on expert knowledge in that the evidential hierarchy needs to be pre-defined and the uncertainty of sensor data needs to be either provided by experts or derived from sensor manufacturers. When the quality of sensors is not well known or there are a tremendous number of sensors, this approach may suffer from incomplete subjective expert knowledge or the requirement of a huge amount of knowledge effort from experts.

In summary, both fuzzy logic and evidence theory that are used in situation identification are applied on top of a specification-based approach. A hierarchy of concepts is needed to use Fuzzy logic to evaluate the similarity contexts and situations, while a hierarchy of sensor data leading to activities is needed to apply Evidence theory in propagating and aggregating beliefs of evidences from sensor data up to activities. Alternatively, these two approaches tackle the uncertainty issue in a quantifiable way while the traditional logic-based approaches cannot, which is described in Equation (1) as shown in Section 3: quantifying the satisfaction degree of conditions in a logical specification of a situation, determining the weight of each condition, and accumulating evidences on each condition.

## 5. Learning-based Techniques

Specifying and identifying situations can have a large variability depending on factors such as time, location, individual users, and working environments [60]. This makes specification-based approaches relying on models of *a priori* knowledge impractical to use. Machine learning techniques have been widely applied to learning complex associations between situations and sensor data. Referring to the information flow as shown in Figure 1 of section 2, the knowledge base between sensor data and inferred situations can be acquired from learning processes, especially correlations of relevant contexts. This section will introduce the main stream of work in learning-based techniques.

### *5.1. Naïve Bayes*

Bayesian classification techniques, including Naive Bayes and Bayesian (belief) networks, are based on Bayes' theorem:

$$P(H \mid X) = \frac{P(X \mid H)P(H)}{P(X)},$$

where $X$ is considered as evidence to support the hypothesis $H$, $P(X)$ and $P(H)$ are the prior probabilities of X and H, and $P(X \mid H)$ is the posterior (conditional) probability of X conditioned on H.

Naive Bayes is a simple classification model that applies Bayes theorem with the strong conditional independence assumption; that is, given the output classification, all the input attributes or features that characterise the evidence are conditionally independent of one another. Considering the above Bayes' theorem, if the evidence $X$ is made up of many attributes, then the computation on the posterior probability $P(X \mid H)$ would be very expensive. The independence premise can reduce the computation by

$$P(X \mid H) = \prod_{k=1}^{n} p(x_k \mid H),$$

where $x_k$ is the value on one of the attributes or features.

Naive Bayes has been used extensively [10, 81, 98, 106, 129, 136]. For example, Muhlenbrock *et al* apply Naive Bayes in learning user activities and user availability. The evidence X is characterised by a set of independent context attributes, including the user's PC usage, phone usage, ambient sound, co-location with other anonymous users, time of the day, and current location. The hypotheses are user activities (e.g., using PC, discussing, or meeting) and user availability (e.g., available for a quick question, or not at all). The estimation of the probabilities in the Naive Bayes depends on the type of attribute values: categorical or continuous. Here since all the attribute values are categorical, the probability parameters are estimated by counting the numbers of occurrences of each activity with different context values. If the attribute values are continuous, other techniques or models (e.g., Gaussian distribution model) need to be applied to characterise the attribute values [51]. The inference process within the Naïve Bayes is straightforward; that is, to choose an activity with the maximum posteriori probability.

In theory, Naïve Bayes have the minimum error rate in comparison to the other classifiers like decision trees and neural network classifiers [51]. In practice, they can also produce comparable classification accuracies, while their performance might degrade when the attribute independence assumption is broken or there is a lack of available probability data. For example, if a certain attribute $x_k$ does not support the hypothesis H in the training data, then the probability $p(x_k \mid H)$ will be zero, which results in the lack of available probability for $x_k$ when it appears in the testing data.

### *5.2. Bayesian Networks*

When there exist dependencies between attributes, Bayesian (belief) networks can be applied to substitute Naive Bayes. A Bayesian network is a directed acyclic graph in which each node represents a variable that can be discrete or continuous, and each arc is the causal relationship between nodes. If there is an arc from a node A to another node B, then A is called a *parent* of B, implying that the variable B is regarded depending directly on A. In this sense, Naive Bayes models can be viewed as Bayesian networks in which each input attribute has an output classification as the sole parent and the classification has no parents.

In a Bayesian network, if a node does not have a parent, then it is called *root*. Each root node is associated with an *a priori* probability. Each non-root node is associated with a conditional probability distribution (CPD). If the variables are discrete, then the CPD is represented with a conditional probability table (CPT) given all possible combinations of their parent nodes: $p(x \mid parent(x))$, where $parent(x)$ is a parent set of a node $x$. To use Bayesian networks, human experts usually apply their understanding of the direct conditional dependencies between attributes to help in the network design. The probability estimation and inference are similar to that in Naive Bayes.

Bayesian networks have been applied in many context-aware systems, such as that proposed by Gu *et al* [44], Ding *et al* [36], Ranganathan *et al* [111], Acellular *et al* [1], Truong *et al* [131], and Ye *et al* [149]. Figure 8 presents a Bayesian network in the work of [44]. The leaf node in this network is the deduced activity node – Tom's current activity, which depends on several contexts, including his location, the living room's lighting level and noise level, the status and location of other people, number of person in his house, his profile (Birthday), etc. In this Bayesian network, the causal relationship exists between the status context of Alice and her location and the status of the micro oven, which breaks the independence assumption in Naive Bayes and thus cannot be modelled in it.

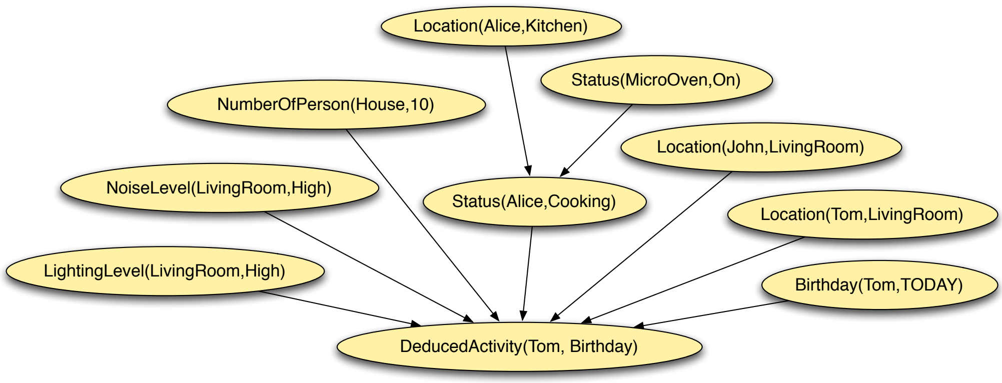


Figure 8: An example of a Bayesian network structure [44]

As presented, Bayesian networks provide a clear and well understood method for incorporating how the likelihood of a possibility of event is conditioned on another event. They are best suited to applications where there is no need to represent ignorance, where conditioning is easy to extract through probabilistic representation and prior odds are available. Similar to Naive Bayes, they may lack credibility due to the unavailability of estimates [57].

### *5.3. Hidden Markov Models*

Hidden Markov Models (HMMs) are a statistical model where a system being modelled is assumed to be a Markov chain that is a sequence of events. As shown in Figure 9, an HMM is composed of a finite set of hidden states (e.g., $s_{t-1}$, $s_t$, and $s_{t+1}$) and observations (e.g., $o_{t-1}$, $o_t$, and $o_{t+1}$) that are generated from states. HMM is built on three assumptions: (1) each state depends only on its immediate predecessor, (2) each observation variable only depends on the current state, and (3) observations are independent from each other [110]. In an HMM, there are three types of probability distributions: (1) priori probabilities over initial states $p(s_i)$, $1 \le i \le n$, where $n$ is the number of states; (2) state transition probabilities $p(s_t \mid s_{t-1})$; and (3) observation emission probabilities $p(o_t \mid s_t)$. The joint probability of a paired observation and state sequences is calculated as:

$$p(\mathbf{s}, \mathbf{o}) = \prod_{t=1}^{T} p(s_t \mid s_{t-1}) p(o_t \mid s_t)$$

where $s_i$ and $o_i$ are a sequence of states and observations respectively.

The training of a HMM is performed using the Baum-Welch algorithm [11, 141]. It is a generalised Expectation Maximisation algorithm that can compute maximum likelihood estimates for the probability parameters for a HMM given the observations as training data [4]. The activity recognition in a HMM is a process of choosing a sequence of states that 'best' explains the input observation sequence, which is performed in the Viterbi algorithm [110].

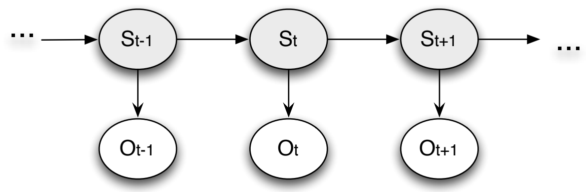


Figure 9: The graphical representation of a HMM [137]

HMMs have been a popular model in the domain of activity recognition [27, 37, 53, 95, 137, 143]. van Kasteren et al. [137] construct a HMM where each state represents a single activity (e.g., 'prepare dinner', 'go to bed', 'take shower', and 'leave house'). They represent observations in three types of characterised sensor data that are generated in each activity, which are raw sensor data, the change of sensor data, the last observed sensor data, and the combination of them. The HMM is training to obtain the three probability parameters, where the priori probability of an activity represents the likelihood of the user starting from this activity; the state transition probabilities represent the likelihood of the user changing from one activity to another; and the observation emission probabilities represent the likelihood of the occurrence of a sensor observation when the user is conducting a certain activity.

Atallah and Yang have given a comprehensive review on standard and derivative HMMs including coupled, hierarchical, and parametric HMMs. Here we will highlight their findings, while more details can be found in [4]. A problem with the use of a standard HMM for activity detection is that due to the first order Markov assumption (a state depending only on the previous one) the probability of an event state being observed for a certain interval of time declines exponentially with the length of the interval. Also the probability that there is a change in the hidden state does not depend on the amount of time that has elapsed since entry into the current state, which could be an important parameter in modelling human activities. To model the priori duration of an event state, Hongeng *et al* augment an HMM to a *semi-HMM*, where the hidden process is semi-Markovian. This semi-HMM performs better at approximating visual primary and composite activities [59].

Another drawback of using a standard HMM is that its lack of hierarchical modelling for representing human activities. To deal with this issue, several other HMM alternatives have been proposed, such as hierarchical and abstract HMMs. Fine *et al* have extended HMMs to a *hierarchical-HMM* (or layered-HMM), which includes a hierarchy of hidden states [38, 101]. Each of the hidden states in this model can be considered as an 'autonomous' probabilistic model on its own; that is, each hidden state is also a hierarchical HMM. Each state generates sequences by a recursive activation of one of the sub-states of a state. The process of the recursive activation ends when a special state (i.e., a *production* state) is reached. A vertical transition in a hierarchical HMM is the activation of a sub-state by an internal state, whereas a horizontal transition refers to a state transition within the same level.

Bui *et al* have extended HMMs to an *abstract-HMM*, who use a hierarchical structure for probabilistic plan recognition [19]. The bottom part of this model consists of hidden states and observations as in a typical HMM. However, the states are linked to abstract policy (or activity) variables which are placed in a hierarchy. Flag variables are used to indicate whether the current policy or activity continues or terminates in the next time step. Abstract HMMs have been used successfully for learning hierarchical models of indoor activity and performed better than flat models. They have also been used to represent and recognise complex behaviours from trajectory data. The underlying problem with both abstract and layered HMMs is that the state space can be very large, augmenting the number of parameters in the transition model.

### *5.4. Conditional Random Fields*

HMMs generally assume that all observations are independent, which could possibly miss long-term trends and complex relationships. CRFs, on the other hand, eliminate the independence assumptions by modelling the conditional probability of the state sequence $p(\mathbf{s} \mid \mathbf{o})$ rather than the joint probability of the states [134]. This allows CRFs to incorporate complex features of the observation sequence $\mathbf{o}$ without violating the independence assumption of observations as required in HMMs.

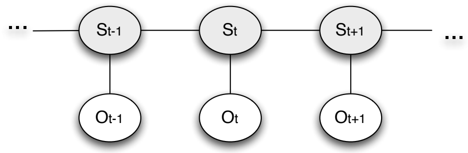


Figure 10: A graphical representation of a linear-chain CRF [137] – note the undirected edges compared to figure 9

CRFs are undirected graphical models, as shown in Figure 10. The conditional probability $p(\mathbf{s} \mid \mathbf{o})$ is computed in terms of cliques and clique potentials rather than the product of conditional distributions as in a directed model. Here, a clique consists of an edge between two adjacent states $s_{t-1}$ and $s_t$ as well as the edge between these two states and the set of observations $\mathbf{o}$ [134]. A clique potential function, $\psi(t, s_{t-1}, s_t, \mathbf{o})$, computes a value analogous to the probability that the variables in its corresponding clique take on a given configuration. The potential function takes the form of $exp(\omega.f(t, s_{t-1}, s_t, \mathbf{o}))$, where $f$ is a feature function that is designed to capture useful domain information and $\omega$ is the weight of the feature function. The conditional probability of a CRF is yielded in the following form:

$$p(s \mid o) = \frac{1}{Z} \prod_{t=1}^{T} exp(\omega.tf(t, s_{t-1}, s_t, o))$$

$$Z = \Sigma_s \prod_{t=1}^{T} exp(\omega.f(t, s_{t-1}, s_t, o)).$$

In the above formula, feature functions are defined by experts and weights need to be learned from training data using an iterative gradient method like Limited Memory BFGS (Broyden-Fletcher-Goldfarb-Shanno) [76]. The inference can be performed similarly to the Viterbi algorithm [137].

A body of work has applied CRFs in activity recognition, including [35, 75, 133, 134, 137]. We take the example from [134] to illustrate the use of CRF by focusing on the definition of feature functions. Vail *et al* apply the CRF in a robot tag application where a seeker robot attempts to tag its closest neighbour robot and transfers the seeker role to the tagged robot. In this CRF model, the observations are a sequence of two-dimensional positions of each robot in an environment, while the states are represented as the ID of the current seeker robot. They define intercept feature functions $f = I(s_t = j)$ (i.e., if $s_t = j$ then $f = 1$, otherwise, $f = 0$) and transition feature functions $f = I(s_{t-1} = i).I(s_t = j)$ as a base model, where $s_t$ indicates the ID of the seeker robot at the time instance $t$. On top of them, specialised feature functions that are transformed from position data are defined, including velocity, distance, and chasing features. For instance, a distance threshold feature function is introduced as

$$f_{r_1,r_2,i,j} = I(s_{t_1} = i).I(s_t = j).I(dist_t(r_1, r_2) \leq k \vee dist_{t-1}(r_1, r_2) \leq k).I(dist_{t_2}(r_1, r_2) > k)$$

where $r_1$ and $r_2$ are positions of two robots, $dist_t(r_1, r_2)$ are the distance between them at the time instant $t$, and $k$ is the distance threshold. This function describes that the change of the seeker role can occur if two robots are within the threshold distance. This example shows how to incorporate of domain knowledge in feature functions; that is, linking characterised observations to state transitions. The use of domain knowledge results in lower error rates in recognising activities than HMMs, as presented in their experiments.

*5.5. (Stochastic) Context Free Grammars*

Context free grammar (CFG) approaches provide a sound theoretical basis for modeling structured processes. A CFG naturally leads the representation to use concepts recursively, enabling the action to be

| Production Rules | | | Description |
|---|---|---|---|
| $S$ | $\rightarrow AB$ | [1.0] | Blackjack → "play game" "determine winner" |
| $A$ | $\rightarrow CD$ | [1.0] | play game → "setup game" "implement strategy" |
| $B$ | $\rightarrow EF$ | [1.0] | determine winner → "evaluate strategy" "cleanup" |
| $C$ | $\rightarrow HI$ | [1.0] | setup game → "place bets" "deal card pairs" |
| $D$ | $\rightarrow GK$ | [1.0] | implement strategy → "player strategy" |
| $E$ | $\rightarrow LKM$ | [0.6] | evaluate strategy → "flip dealer down-card" "dealer hits" "flip player down-card" |
| | $\rightarrow LM$ | [0.4] | evaluate strategy → "flip dealer down-card" "flip player down-card" |
| $F$ | $\rightarrow NO$ | [0.5] | cleanup → "settle bet" "recover card" |
| | $\rightarrow ON$ | [0.5] | → "recover card" "settle bet" |
| $G$ | $\rightarrow J$ | [0.8] | player strategy → "Basic Strategy" |
| | $\rightarrow Hf$ | [0.1] | → "Splitting Pair" |
| | $\rightarrow bfffH$ | [0.1] | → "Doubling Down" |
| $H$ | $\rightarrow l$ | [0.5] | place bets |
| | $\rightarrow lH$ | [0.5] | |
| $I$ | $\rightarrow ffI$ | [0.5] | deal card pairs |
| | $\rightarrow ee$ | [0.5] | |
| $J$ | $\rightarrow f$ | [0.8] | Basic strategy |
| | $\rightarrow fJ$ | [0.2] | |
| $K$ | $\rightarrow e$ | [0.6] | house hits |
| | $\rightarrow eK$ | [0.4] | |
| $L$ | $\rightarrow ae$ | [1.0] | Dealer downcard |
| $M$ | $\rightarrow dh$ | [1.0] | Player downcard |
| $N$ | $\rightarrow k$ | [0.16] | settle bet |
| | $\rightarrow kN$ | [0.16] | |
| | $\rightarrow j$ | [0.16] | |
| | $\rightarrow jN$ | [0.16] | |
| | $\rightarrow i$ | [0.18] | |
| | $\rightarrow iN$ | [0.18] | |
| $O$ | $\rightarrow a$ | [0.25] | recover card |
| | $\rightarrow aO$ | [0.25] | |
| | $\rightarrow b$ | [0.25] | |
| | $\rightarrow bO$ | [0.25] | |

| Symbol | Domain-Specific Events (Terminals) |
|---|---|
| $a$ | dealer removed card from house |
| $b$ | dealer removed card from player |
| $c$ | player removed card from house |
| $d$ | player removed card from player |
| $e$ | dealer added card to house |
| $f$ | dealer dealt card to player |
| $g$ | player added card to house |
| $h$ | player added card to player |
| $i$ | dealer removed chip |
| $j$ | player removed chip |
| $k$ | dealer pays player chip |
| $l$ | player bets chip |

Figure 11: A SCFG for the Blackjack card game [96]

defined based on sub-events. Ryoo *et al* propose a CFG based approach to construct a concrete representation for any composite action, where atomic actions (e.g., human poses or gestures like 'stretch' or 'withdraw') serve as terminals and composite actions (e.g., 'shake hands') serve as non-terminals. The non-terminals can be converted to terminals recursively through production rules.

Stochastic context free grammars (SCFGs) are a probabilistic extension of CFGs, where production rules are augmented with probabilities that provide a quantitative basis for ranking and pruning parses as well as for exploiting dependencies in a language model [96]. They have been applied to model the semantics of activities whose structure is assumed to be known in advance [61, 96]. For example, Moore *et al* use a SCFG to recognise separable, multitasked activities, which are defined as 'the intermittent co-occurrence of events involving multiple people, objects, and actions, typically over an extended period of time' [96]. Take the Blackjack card game as an example. They construct a SCFG by enumerating a list of primitive events that need to be detected (e.g., 'player bets chip' or 'dealer added card to house') and a set of production rules that define the higher activities of interest (e.g., 'play game' or 'evaluate strategy'), as shown in Figure 11. A modest amount of training data is required to obtain the probabilities on the production rules. The *Earley-Stolcke* algorithm is employed to parse input strings and accommodate the SCFG. This algorithm uses a top-down parsing approach and context free productions to build strings that are derivable from left to right. It maintains multiple hypotheses of all possible derivations that are consistent with the input string up to a certain point. Scanning the input strings from left to right, the number of hypotheses increases as new options become available or decreases as ambiguities are resolved. The evaluation results show that the SCFG performs well in detecting higher level activities and as well as in dealing with the errors in the low-level tasks such as tracking errors and missing observations.

Both CFGs and SCFGs work well when the structure of the activities of interest is not very complex and is well known to developers. In complex scenarios involving several agents requiring temporal relations that are more complex than just sequencing, such as parallelism, overlap, synchrony, it is difficult to formulate the grammatical rules manually. Learning the rule of the grammar from training data is a promising alternative, but it has proved to be extremely difficult in the general case [132].

### 5.6. *Decision Trees*

A decision tree is a predictive model where each leaf represents a classification and each branch represents a conjunction of features that lead to the target classifications. A decision tree is built on information entropy in that its construction works by choosing a variables at each step that is the next best variable to use in splitting the set of items. Compared to Bayesian models, a decision tree does not take into account the dependence assumption or potential temporal sequence between classifications.

One of the main advantages of using a decision tree is that it can generate classification rules that are easy to understand and explain classification rules. These rules are useful in analysing sensor performances and feature extraction [10]. [10, 25, 66, 85] have used decision trees in classifying human activities from the body acceleration data.

Traditional decision trees like C4.5 are efficient in reasoning for relatively small data sets. However, the efficiency in constructing them is subject to the size of training data, since the construction of these trees require that the training data reside in memory. Therefore, when they are applied to mining a very large real-world data sets (millions of entries), they will become inefficient due to swapping of the training data in and out of main and cache memories [51].

### 5.7. *Neural Networks*

(Artificial) Neural networks are a sub-symbolic technique, originally inspired by biological neuron networks. They provide an alternative technique for activity recognition. They can automatically learn the complex mappings and extract a non-linear combination of features. A neural network is composed of many artificial neurons that are linked together according to a specific network architecture. Figure 12 shows a structure of a neural classifier, which consists of an input layer (e.g., $u_1$, $u_2$, $\ldots$, $u_r$), a hidden layer, and an output layer (e.g., $y_1$, $\ldots$, $y_h$). Mappings between input and output features are represented in the composition of activation functions $f$ at a hidden layer [99], which can be learned through a training process. Yang *et al* have discussed well known learning algorithms, including a gradient descent optimisation method [52, 107] and resilient backprogagation algorithms [116].

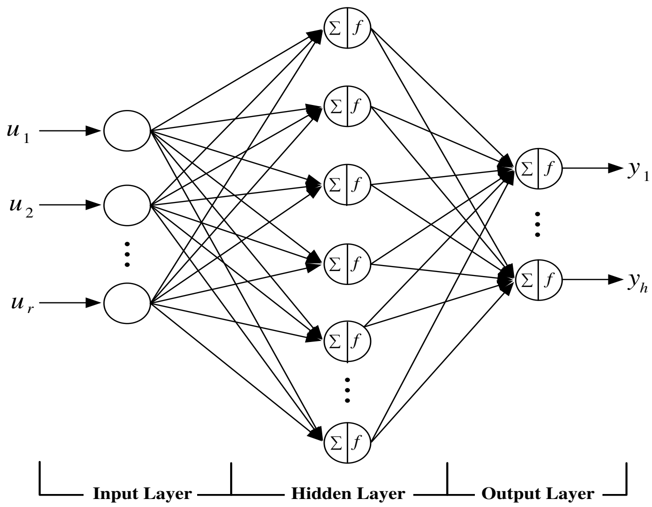


Figure 12: A structure of a neural classifier [145]

Yang et al. [145] employ neural networks in learning human activities (e.g., static activities like 'sitting' and 'working at a PC', and dynamic activities like 'running' and 'vacuuming') from acceleration data. The acceleration data is collected from a wireless sensing triaxial accelerometer module mounted on the dominant wrist of users. Eight features are extracted from the acceleration data, including the mean value, the correlation between axes, the energy that is used to discriminate sedentary activities from moderate and vigourous ones, and so on. With three axes, they have 24 features in total, which is considered a large number of features that will make the computation speed and the training process difficult. Also considering that some of the features might be redundant, irrelevant or insignificant to the improvement of recognition accuracy, they choose a subset of the features, which is based on the common principal component analysis [39, 68, 153] and the k-means clustering technique. This subset of features are used to construct neural networks for static and dynamic activities respectively. After training with the paired input features and output activities, they are ready to be used in recognising activities.

The performance of neural networks are affected heavily by the amount of training data [99]. A neural network is considered a good choice if there is plenty of training data and the problem domain is poorly understood to derive an approximate model. On the contrary, if the training data does not cover a significant portion of the operating conditions or if the data is noisy, then neural networks will not perform well.

### *5.8. Support Vector Machines*

Support Vector Machines (SVM) are a relatively new method for classifying both linear and nonlinear data. A SVM uses a nonlinear mapping to transform the original training data into a higher dimension. Within this new dimension, it searches for the linear optimal separating hyperplane that separates the training data of one class from another. With an appropriate nonlinear mapping to a sufficiently high dimension, data from two classes can always be separated. SVMs are good at handling large feature spaces since they employ overfitting protection, which does not necessarily depend on the number of features [104]. SVMs have been applied in activity recognition by [65, 104]. Kanda et al use SVMs to categorise human motion trajectories (such as 'fast walk', 'idle walk', 'wander', and 'stop') based on their velocity, direction, and shape features.

### *5.9. Web Mining*

Considering the engineering effort and economical cost in setting up a pervasive computing environment, training data can be critical or unavailable, which will eliminate the option to use a machine learning technique. To explore the association between sensor data and situations, recent researchers [43, 103, 109, 128] have applied the web mining techniques to gain this type of knowledge.

Palmes *et al* use the web mining techniques to extract key terms for each targeted activity from its collected online 'howto' documents, and to determine each term relevance weight for each object. The assumption underlying this approach is that in most activities of daily living, the lists of relevant objects for a particular activity are similar and do not vary significantly even when the activity is performed in different ways. Their results show that the topmost term has the highest relevance for each particular activity. Also the topmost terms are unique among all the activity models. These two observations form the foundation of their approach; that is, the topmost terms are used as the key objects to discriminate various activities. Based on the relevance of objects to an activity, instead of the order of objects, they have applied two heuristic algorithms *MaxGap* and *Gain* to support automatically detecting and segmenting sensor traces.

This approach can work well for non-interleaving activities in a pre-determined list. However, its performance can be undermined (1) if the activities share common key objects; or (2) if the access of objects key to a certain activity has not been sensed, which could be that key objects have not been associated with sensors or the sensors on key objects perform poorly.

### *5.10. Suffix Trees*

Hamid *et al* present an unsupervised framework to discover temporal patterns of finer-grained events in everyday human activities [49, 50]. Here an activity is defined as a finite sequence of events, while an event is defined as a particular interaction among a subset of key objects over a finite duration of time. A key object is an object present in an environment that provides functionalities that may be required for the execution of an activity of interest [50]. The idea is to extract events from interactions between human users and key objects, organise events into a Suffix-Tree, and mine the patterns of events in each activity. A Suffix-Tree enumerates all unique subsequences of events in a certain activity, where any subsequence of events from the root node to a non-root node is called a *event motif*. For example, Figure 13 presents a Suffix-Tree, where one of the event motifs is {1, 2, 3, 2, 3, 2, 4}, representing 'Fridge → Stove → Table → Stove → Table → Stove → Sink'. Here the events are simply represented as as the name of key objects such as 'Fridge' and 'Stove'.

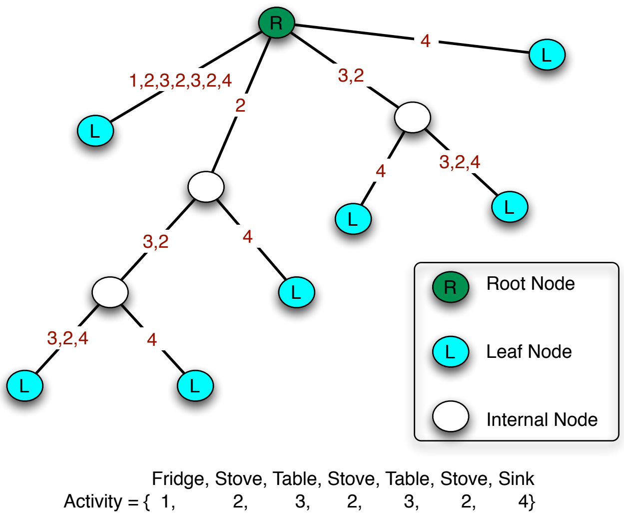


Figure 13: An example of a Suffix Tree [49]

The activity class discovery method is clustering-based. A completely connected edge-weighted activity graph is generated, where each node represents an activity instance in a training data set and each edge represents the similarity between the two connected nodes. The discovery of activity classes is to search for edge-weighted maximal cliques in the activity graph. The activity recognition process is done by evaluating the similarity between a given new activity instant and each activity class; that is, calculating the average similarity degree between this new instance and each activity instance in an activity class and the result activity class is the one with the highest degree.

The main contribution of this work is an investigation of knowledge representation and manipulation techniques that can facilitate learning of everyday human activities in a minimally supervised manner. The usage of Suffix-Trees provides an efficient activity representation capable of analysing event sequences over the entire continuum of their temporal scale. In this approach, event motifs partially capture the global structural information of activities. The probabilistic occurrence of event motifs makes it scalable to environments with sensor noise [49]. However, the way that it rigidly models an activity sequence in event motifs makes it sensitive to the order of the events [47] and unable to handle an activity with interleaving or overlapping events.

### *5.11. Emerging Patterns*

An Emerging Pattern (EP) is a type of knowledge pattern that describes significant changes between two classes of data. An EP is a set of items whose frequency changes significantly from one data set to another. Given two different classes of data sets $D_1$ and $D_2$, the growth rate of an itemset $X$ from $D_1$ to $D_2$ is defined as

$$GrowthRate(X) = \begin{cases} 0 & \text{if } supp_1(X) = 0 \text{ and } supp_2(X) = 0 \\ 1 & \text{if } supp_1(X) = 0 \text{ and } supp_2(X) > 0 \\ \frac{supp_2(X)}{supp_1(X)} & \text{otherwise} \end{cases}$$

where $supp_i(X)$ $(i = 1, 2)$ is the support of an itemset $X$ in a data set; i.e., $count_i(X)/|D_i|$, $count_i(X)$ is the number of instances in $D_i$ containing $X$. EPs are the itemsets with large growth rates from $D_1$ and $D_2$.

Gu *et al* propose a novel Emerging Patterns based approach to recognise sequential, interleaved and concurrent activities [43, 47, 48, 138]. For each sequential activity, they collect the instances in the training data set, while each instance refers to a union of all the sensor observations that belong to the sequential activity during a continuous period of time. Then they mine a set of EPs for each sequential activity using the algorithm in [74]. Each EP consists of a collection of characteristic sensor observations; for example, for the 'cleaning a dining table' activity, one of the EPs is 'location@kitchen, object@plate, object@cleanser, object@wash_cloth', with a support of 95.24% and a growth rate $\infty$. It means that the cleanser, plate, and wash_cloth are the common objects which are involved in this activity, and this activity usually occurs in the kitchen.

To detect a correct activity for a current observation sequence at a time $t$, they first obtain test instances for each possible activity $A_i$, each of which computes the union of all the sensor observation from $t$ to $t+L_{A_i}$ where $L_{A_i}$ is the average duration of an activity $A_i$. Then they apply the test instances in a scoring function and choose the activity with the highest score. The scoring function is

$$score(A_i, A_j, S_{t\ \ t+L_{A_i}} = c_1.ep\_score(A_i, S_{t\ \ t+L_{A_i}}) + c_2.coverage\_score(A_i, S_{t\ \ t+L_{A_i}}) + c_3.P(A_i|A_j),$$

where $A_j$ is the activity ends at the time $t$ and $S_{t\ \ t+L_{A_i}}$ is the test instance generated for a current candidate activity $A_i$. In this scoring function, the sub function *ep_score* computes the degree to which the test instance is supported by the EPs in the activity $A_i$, *coverage_score* measures the fraction of irrelevant observations in a sliding window, and the $P(A_i|A_j)$ is the activity correlation score; that is, the conditional probability of the activity $A_i$ given that $A_j$ occurs before.

### *5.12. Intertransaction Association Rule Minings*

Another data mining technique similar to the above EP is Intertransaction Association Rule (IAR). EPs have the ability to identify the discriminative features between data sets, while IAR mining is concerned with the discovery of sets of items that frequently occur together within the records of a transactional database [82].

Lühr *et al* apply IAR mining to detect abnormal behaviour of occupants in smart home environments. The IARs allow the one to capture the associative, non-sequential, relationship of events while retaining the higher level temporal context in which these events occur. One of the IAR examples is

$kitchen\_sink\_cold\_water\_on_0, kitchen\_sink\_cold\_water\_off_0, dishwasher\_open_1, dishwasher\_closed_3 \Rightarrow dishwasher\_on_3$.

In this rule, the sensor events $kitchen\_sink\_cold\_water\_on$ and $kitchen\_sink\_cold\_water\_off$ are in the current transaction interval, $dishwasher\_open$ in the next, and $dishwasher\_closed$ and $dishwasher\_on$ three transactions from now. It implies that the occupant opens the dishwasher, closes it again and then turns the machine on shortly after having used the cold water faucet in the kitchen sink, which could be a pattern of the occupant rinsing the dishes.

To gain insight into an occupant's behaviour, they introduce the use of emergent IARs that display significant growth rate from one data set to another. Their presence may indicate abnormality evident as either a previously unseen pattern of events or unusually frequent occurrences of behaviour that would otherwise be considered normal. They also provide a means to distilling a large number of discovered rules down to those most useful or interesting ones; for example, 7200 patterns can be distilled down to 150 emergent rules [82].

### *5.13. Jeffrey Divergence*

Jeffrey divergence can be used to separate different observation distributions by calculating the divergence between the histograms of two adjacent observation windows [17, 18]. Brdiczka *et al* propose an unsupervised method based on the Jeffrey divergence to detect small group meeting activities (such as 'discussion', 'presentation', or 'questions') from a stream of multimodal observations; that is, audio and video information of participants in a group meeting, which are recorded with microphones and cameras. The proposed method first generate histograms of observations from adjacent windows of variable size slid from the beginning to the end of a meeting recording. The peaks of Jeffrey divergence curves between the adjacent histograms can be used to detect distinct distributions of observations, which can be interpreted as distinct segments of group activities.

The Jeffrey divergence based method is completely unsupervised, which is advantegeous when it is applied to analyse group activities with an increasing number of participants and pre-unknown activities. This method provides the first segmentation of small activities in a meeting. These detected segments can be then used as input for learning and recognising coarser-grained meeting situations such as a small group meeting or a seminar.

## 6. Discussion

The previous sections have introduced the main stream of situation identification techniques in details. In this section, we will provide a qualitative evaluation on them in terms of situation identification at different levels of abstraction, uncertainty, temporality, knowledge incorporation and derivation, and engineering effort.

### *6.1. Summary of Techniques*

Compared to the specification-based approaches, one of the most distinguishable features of the learning-based approaches is their ability in uncovering a pattern or correlation underlying data. All these learning-based approaches can be used to extract categorical features from numeric sensor data; for example, learning movement or motions of human body such as running or walking from acceleration data. They can learn correlations between a combination of relevant categorical or numeric sensor data) and situations; for example, learning the pattern of how users interacting with sensorised objects when they perform a certain activity.

| **Learning Techniques** | **Supervised** | **Unsupervised** | **Static** | **Sequential** |
|---|---|---|---|---|
| Naïve Bayes | √ | | √ | |
| Bayesian Networks | √ | | √ | |
| Hidden Markov Models | √ | | | √ |
| Conditional Random Fields | √ | | | √ |
| Context Free Grammars | √ | | | √ |
| Decision Trees | √ | | √ | |
| Neural Networks | | √ | √ | |
| Support Vector Machines | √ | | √ | |
| Web Mining | | √ | √ | |
| Suffix-Trees | | √ | | √ |
| Emerging Patterns | √ | | √ | |
| Intertransaction Association Rule Mining | √ | | | √ |
| Jeffrey Divergence | | √ | √ | |

Table 2: Supervised vs. unsupervised and static- vs. sequential-based learning

Most the examined techniques are supervised learning as presented in Table 2, such as Naive Bayes, Bayesian networks, HMMs, CRFs, and so on. These techniques learn the models and parameters from training data that usually requires human to label a situation to sensor data that are observed during the occurrence of this situation. When there exists a large number of situations to be identified in our daily lives, manual labeling of training data may place a significant burden to developers involved in the data collection. Therefore, supervised learning techniques may have limitations in real-life deployment where scalability, applicability, privacy, and adaptability are highly concerned [43]. To tackle this issue, researchers have employed unsupervised learning approaches. Among them, neural network, suffix-tree, and Jeffrey divergence can extract features from sensor observations, which are distinguishable from one situation to another. Web mining techniques are not strictly unsupervised learning in that they perform the learning on web documents, rather than on the collected sensor data. The learning results can be directly used to infer human activities [103] or as a source for other supervised learning techniques [43].

Also these learning techniques can be classified into static- and sequential-based as presented in table 2. Static-based approaches aim to capture the features of sensor observations during the occurrence of a situation; for example, a Naive Bayes model represents how the different characteristic values of each sensor observation contribute to identifying a certain situation. In contrast, sequential-based approaches aim to capture the temporal relationships between sensor observations and those between situations. For example, a HMM incorporates the sequential order between situations (i.e, $p(s_t|s_{t-1})$ in section 5.3) in the model; IARs allow to capture the higher-level temporal context in which associative sensor events occur [82]; that is, elapsing time units between the occurrence of the events. The fact that human activities usually follow a certain sequential pattern makes the sequential-based approaches more amenable to real world use than static-based approaches.

*6.2. Situation Identification at Different levels of Abstraction*

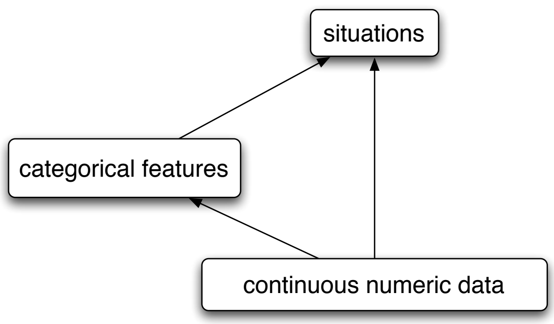


Figure 14: Situation identification at different levels of abstraction

As discussed in section 2.1, different types of sensor data lead to different techniques to analyse them. Figure 14 shows situation identifications at different levels of abstraction. First of all, situations can be recognised directly from continuous numeric data; for example, inferring human motions like ‘walking’ or ‘running’ from acceleration data. Situation identification at this level is usually performed in learning-based approaches, which uncover complicated associations (e.g., nonlinear) between continuous numeric data and situations by carving up ranges of numeric data (e.g., decision tree) or finding an appropriate algebraic function to satisfy or ‘explain’ data (e.g., neural networks or SVMs). Compared to other learning-based approaches, Bayesian networks, HMMs, CRFs are less capable in learning the patterns of numeric data in that the parameters in these models, i.e., evidence or observation vectors, are more amenable to categorical features. Specification-based approaches can apply if the association between sensor data and situations are rather explicit and representable in logic rules.

Situations can also recognised from categorical features; for example, inferring a room’s situation – ‘meeting’ or ‘presentation’– from the number of persons co-located in the room and the applications running in the computer installed in the room. This higher-level of situation identification can be performed in both specification- and learning-based approaches. As stated in section 3, when (1) there exists a relatively smaller size of sensors, (2) sensor data are relatively accurate, and (3) domain experts have a good understanding of situations, experts can define logical specifications of situations into logic rules, and the corresponding inference engines can be applied to perform reasoning. Otherwise, learning-based approaches are needed to explore the combination pattern of categorical features of sensor data and situations. In addition, similar to the above low-level of situation identification, a learning-based approach also applies when there is a need to extract categorical features from continuous numeric data; for example, inferring human activities like ‘watching TV’ or ‘cooking’ from a combination of numeric sensor data like the usage of electrical current and gas flow.

### 6.3. Uncertainty

Dealing with uncertain sensor data is one of the main research issues in situation identification. Specification-based approaches introduce uncertainty metrics to describe sensor data, including incompleteness, accuracy, timeliness, and reliability [28, 42, 54, 71]. The concept hierarchy in ontologies are typically used to evaluate the precision of sensor data against the conditions of rules, as mentioned in section 4.

Uncertainty can also exist in the use of crude or oversimplified rules that are defined in an *ad hoc* way [54, 119]. In representing uncertainty of rules, OWL can be extended with conditional probabilistic class to encode the probability that an instance belongs to a class respectively given that it belongs to another class [46, 111, 131]. Although good at expressing uncertainty, these qualitative approaches need to be combined with other techniques like Fuzzy Logic and Evidence theory to quantify the uncertainty to be used in situation identification. Table 3 lists qualitative approaches in resolving uncertainty. Fuzzy logic can deal with imprecision through conceptual matching with the help of a pre-defined structure between concepts and membership functions. A temporal decaying function can be defined as a fuzzy membership function to discount the confidence of sensor data with time [108, 112]. Evidence theory can deal with missing sensor data by assigning incompleteness to uncertainty and combine multiple contradictory sensor sources with a series of mathematical functions.

Learning-based approaches have stronger capability to resolve uncertainty by training with the real-world data that involves noise. These approaches not only learn associations between sensor data and situations, but also the effect that the uncertainty of sensor data has on the associations. For example, the conditional probabilities learned in Naive Bayes includes the reliability of sensor data as well as the contribution of the characterised sensor data in identifying a situation. However, the incompleteness of sensor data can undermine the performance of Bayesian approaches including Naive Bayes, Bayesian Networks, and HMMs: (1) if the sensor data is missing in the training data, then these models will suffer from the lack of available probabilities in the process of inference; (2) if the sensor data is missing in the testing data, then still the corresponding probability could be 0. Information-based techniques like Decision trees, neural networks, and SVMs are less resistant to imperfect sensor data, which are more useful in learning the model. Training them with noisy data could lead to the *overfitting* issue. However, decision trees can deal with overfitting with a post-prunning method, and they, especially C4.5, have been improved with the capability of dealing with missing values and noisy data including inaccurate and contradictory data [142]. SCFGs can resist to low-level sensor noise to a certain degree; for example of the production rules in Figure 11, given that the first input string is 'a', the possible hypotheses will be '$0.A \rightarrow a.a$' and '$0.A \rightarrow a.a.A$', so the expected next input string would be 'a' again. If the next input string is 'b', then the SCFG will automatically ignore this incorrect string and wait for the next one or terminate as a failed parsing.

| Techniques | Incomplete | Imprecise | Inaccurate | Contradictory | Out of date |
|---|---|---|---|---|---|
| Fuzzy Logic | | √ | | | √ |
| Evidence Theory | √ | | | √ | |
| Bayesian Model | | | √ | √ | |
| Conditional Random Field | | | √ | √ | √ |
| Decision Tree | √ | | √ | √ | |
| Stochastic Context Free Grammar | √ | | √ | √ | |

Table 3: Techniques in resolving uncertainty

### 6.4. Temporality

Temporal implication on situations can provide important clues on the types of current situations [29]. There are three research issues on temporal sequences: (1) how to mine sequential pattern; (2) how to use temporal features including the mined sequential pattern in reasoning; and (3) how to detect the boundary where situations change. In the first research issue, a simple way is to locate each pair of events that are temporally adjacent; that is, what has happened immediately before or after an event [137]. In a further advance, a state-of-the-art sequential pattern mining algorithm – Apriori [2] – has been used to discover frequently-occurring events in certain sequences [62, 63]. For example [62], given two sample sequences X = '**AA** BB PP **BB** ZZ **CC** KK **DD** UU VV **EE**' and Y = '**AA BB CC** GG **DD EE**', a sequential pattern can be mined from them: **AA BB CC DD EE**. Aztiria *et al* [7, 8] develop a system called *SPUBS* – Sequential Patterns of User Behaviour, which discovers users' common behaviours and habits from data recorded by sensors. They first identify the sets of actions that frequently occur together, such as 'Cupboard on', 'Cupboard off', 'Cup on', and 'Fridge on'. Then they discover the topology among these frequent sets of actions; for example, an action can occur before another action, certain actions can repetitively occurs in an activity, and simultaneous actions occur unorderly in an activity. Beyond these qualitative sequential relationships, SPUBS also supports mining the quantitative relationships; for example, a user turns on the light of the bathroom 4 seconds after he enters it. Choujaa and Dulay [26] use *mutual information* [123] to measure the long-range dependencies between events; that is, a user's current behaviour might be related to his past activities at remoter times of the day.

In applying temporal features in situation identification, Ye et al. [147] have demonstrated the potential for applying temporal features in recognising human activities in a smart home environment. The *time of day* temporal feature has been used to semantically interpret timestamps on sensor data; The *duration* feature has been used to collect sensor data in a certain period so as to infer an activity [90, 129]. Specification-based approaches like [5] model the temporal relationships that are acquired from expert knowledge in rules, where the operators to define temporal relationships are borrowed from the work in the domain of natural language processing [40]. [114] model the sequences of situations of an automatic cameraman in Allen's temporal logic in Figure 3, which have been compiled into a synchronised Petri net for inference; for example, a situation 'Starts recording' `meets` another situation 'Lecturer speaks', and 'Lecturer speaks' is `during` 'Someone enters'. Learning-based approaches like HMMs model the transition between situations as described in Section 5.3; that is, the identification of the current situation depends on its previous situation. However, they might not be suitable for handling temporal sequences if they are not immediately adjacent, as introduced in the above paragraph.

Detecting a boundary where a situation changes to another situation is also an important research issue, which can be used to prevent hazardous situations and take in time actions; for example, detecting a user from an exercise situation 'jogging' to a dangerous situation 'fainting'. [103] detect the boundary of human daily activities by discriminating the difference between the weight of objects in two adjacent activities. [4] suggest that correct recognition of transitional activities and observing inter-subject and intra-subject differences would offer a clearer idea of situation variation.

### *6.5. Complexity of Situations*

The current situation identification techniques have moved from recognising simple sequential situations to more situations with more complicated structural features, such as interleaved activities with multiple users [43, 96, 105, 139].

Activities involving multiple users collaboratively or concurrently are common in our daily lives. People often form groups to perform certain activity collectively so that they can interact and rely on each other to achieve specific goals [139]; for example, couples cooking together in a kitchen, or several users performing pipeline tasks. However, recognising activities of multiple users is far more challenging than recognising simple sequential activities. Wang *et al* propose a Coupled Hidden Markov Model (CHMM) to recognise multi-user activities from sensor readings in a smart home environment, as shown in Figure 15. This two-chain CHMMs are constructed by bridging hidden states of the two component HMMs with the crosswork of conditional probabilities $P_{a_t|a_{t-1}}$ and $P_{b_t|a_{t-1}}$. These HMMs represent two sequences of states A and B for two different users, with the observations $O^a$ and $O^b$ respectively. Given an observation sequence $O$, the inference process is to find a state sequence $S$ that maximises the probability $P(S|O)$, where $S$ involves two state sequences $S_a$ and $S_b$ corresponding to the recognised activity sequence for the two users.

Also Patterson *et al* apply HMMs to recognise cooking activities during a morning household routine in the presence of interleaved and interrupted activities. Moore *et al* use SCFGs to model activities that have several independent threads of execution with intermittent dependent interactions with each other, as described in Section 5.5. Gu *et al* build a multi-user model based on the emerging pattern mining approach.

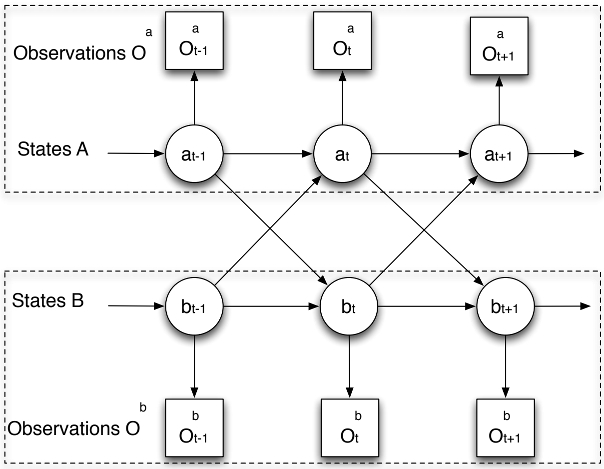


Figure 15: A Coupled Hidden Markov Model to recognise multi-user activities [139]

In this model, an activity sequence for each user is mined separately, while an *interaction score* function is introduced to calculate the probability of one user performing an activity based on the immediately previous activities of all the other users.

### *6.6. Knowledge Incorporation and Derivation*

The descriptions in Section 4 and 5 show that domain knowledge is important in both specification- and learning-based approaches, even though expert knowledge can be biased, inaccurate, or defined in an *ad hoc* way. [150] argue that experts can have more accurate knowledge in local domains while less in combining knowledge in local domains into the whole domain. That is, experts can have a better understanding on sensors, environments, and user preferences, which can be called *local domain knowledge*, so they can define individual concepts in local domains (e.g., symbolic location contexts on raw positioning sensor data) and specify semantic relationships between them (e.g., spatial relationships). It is more difficult for experts to uncover the relevancy of sensor data and their contribution to a certain situation, which can be called *whole domain knowledge*.

As stated, specification-based approaches are developed on expert knowledge, including the local and whole domain knowledge. Learning-based approaches can benefit if the more accurate local domain knowledge applies. As demonstrated in Section 5, models of Bayesian networks can be constructed by experts [44]; CRFs improve the recognition accuracy by incorporating into feature functions the domain knowledge on robot seeking applications, such as distance between robots and their moving velocity. Also Learning-based approaches can complement the inaccuracy of expert knowledge by training through real-world data so that they can be used to uncover unknown relationships and personalise situations' specifications given that a situation's specification can vary with time, location, and individuals.

Incorporating domain knowledge can improve the accuracy of identifying situations, while the capability of deriving knowledge will facilitate the share and reuse of knowledge in other similar environments and applications. Ontologies are a promising technique of representing knowledge to support the share and reuse. Some learning-based approaches can derive knowledge; for example, decision trees can derive a set of association rules, while the others can be difficult, especially neural networks and SVMs.

### *6.7. Knowledge Engineering Effort*

Knowledge engineering effort indicates how much effort developers need to devote to using an approach, which includes (1) extracting features of sensor data based on the understanding of sensors, (2) abstracting domain knowledge on environment and users; (3) defining targeted situations (i.e., association between sensor data, domain knowledge, and situations); and (4) specifying relationships between situations, including composition, dependency, and temporal sequence.

Although to varying degrees, knowledge effort is required from developers in most of the approaches. Specification-based approaches are built heavily on expert knowledge, while a significant amount of knowledge is also required in learning-based approaches as introduced in Section 5. [45] build a structure of a Bayesian network based on their understanding of relationships between sensor data and situations as shown in Figure 8, and learn the probabilities from training data; and [134] apply domain knowledge in extracting features from sensor data as input to the conditional random field models.

Learning-based approaches, especially those introduced in this paper, are well established in the AI area, so they are well supported by mature algorithms or tools. For example, *Weka* is a machine learning software that collects a set of state-of-the-art classification algorithms [142], among which C4.5 decision trees, Naive Bayes, and SVMs are most popularly used by researchers [10, 14, 77, 86, 150]. However, learning-based approaches usually need a large amount of training data to build structures and estimate parameters, while training data is scarce in pervasive computing environments due to the privacy violation issue and the difficulty of recording and annotating activities. Therefore, it is a challenging problem to balance the use of expert knowledge and the required amount of training data.

### *6.8. Sensors in Situation Identification*

Collecting human activities or situations from an environment is important to evaluate situation identification techniques. Section 2.1 has introduced main types of sensors in situation identification. Cameras are traditional sensors in the area of computer vision, from which image features can be extracted from videos and then be applied to inferring high-level human activities.

At the present, positioning sensors, object interaction sensors and acceleration sensors are three of the popular types of sensors in pervasive computing. Location can be an important factor for suggesting a user's activity; for example in common-sense knowledge, cooking in a kitchen, hygiene in a bathroom, watching TV in a living room, and sleeping in a bedroom [77]. Location alone cannot pinpoint the exact activities, but it is a key index for narrowing down possible activities a user being doing and querying other sensor data about a user's or environment [126]. For example, if a user is sensed in a kitchen, a system can then collect data from the sensors installed in the kitchen to further decide what cooking-relevant activity is going on [151], such as 'making coffee', 'cooking a meal', or 'washing dishes'.

The object interaction sensors include RFIDs, binary-state sensors that detect if an associated object is accessed or not [137], and object motion sensors that sense motion of installed objects [77]. These sensors can provide direction information about user activities; for example, if a RFID tag on a kettle is scanned by the reader on the user's wrist, the user is probably boiling the water [103].

Acceleration sensors can be installed on devices such as a mobile phone or worn on a human body such as arms or thighs, and they are used to detect users' movement, including walking, cycling, taking lift, and etc [10, 20, 22, 86, 145]. These sensors usually need to be combined with other types of sensors so as to be able to infer higher-level user activities [16, 47].

Other types of sensors including resource usage sensors and ambient sensors that monitor sound, temperature, or humidity can be used to directly infer location- or object-centered situations; for example, the high sound level in a room can suggest that there a discussion is going on. They also can provide indirect information about user activities; for example, the electricity current reading can indicate that a TV is on, which might suggest that a user is watching it [151].

In summary, different types of sensors provide information of different natures that can be used for identifying different situations [9]. With the advanced development of sensing technologies, it is more likely to set up various types of sensors in an environment or on a user, which makes a situation identification technique that accommodates multiple sensors more favourable in the future.

### *6.9. Summary*

To summarise, specification techniques typically require hand-crafting of situation specifications and inference rules, with heavy reliance on domain knowledge. Reasoning decisions are usually transparent, allowing users to develop a mental model of the system and thus understand system decisions. The incorporation of temporal knowledge and the management of uncertainty can be achieved in some cases; for example, temporal logic or evidence theory for uncertainty, but the selection of a particular specification technique should be based on a deep understanding of its particular strengths or weaknesses. Learning techniques, on the other hand, remove the reliance on domain knowledge but typically require extensive training data from the target environment in order to develop the learning model. Reasoning is hidden from the user to varying degrees depending upon the learning technique, thus challenging the user's ability to scrutinise decisions. Uncertainty is implicitly catered for, a key strength of machine learning approaches. Some learning approaches inherently cater for dynamic sequencing (such as HMMs and CRFs) thus supporting temporality.

## 7. Open Research Opportunities

We will conclude by reviewing some of the questions that seem to remain unresolved in the literature of situation identification.

### *7.1. A Hybrid Approach*

In our view, a combination of specification- and learning-based approaches is required to support successful situation identification in a variety of environments and scenarios. Specification-based approaches provide the ability to represent situations and incorporate the rich knowledge and semantics required to reason about them. As new sensors and new situations emerge, the representations can extended in a controlled knowledge-driven way. Learning approaches, on the other hand, have the ability to analyse raw data, and can thus extract patterns and deal with uncertainty.

A hybrid approach will take the 'best of both worlds' by utilising a combination of techniques. Such an approach will be able to provide a formal, semantic, and extensible model with the capability of dealing with uncertainty of sensor data and reasoning rules. More specifically, a desirable situation model is built on a solid logical system that incorporates logical primitives rich enough to represent all sorts of information including sensor data, abstracted context, domain knowledge, and situational knowledge, and rules to check consistency and integrity of the knowledge base. Ontologies have been and will still be a preferred choice to translate, represent, and instantiate the knowledge. A situation model can be plugged with different learning-based techniques, which can be used to derive situation specifications and infer situations when input with noisy sensor data.

### *7.2. Application-led reasoning*

From the perspective of applications, it is insufficient to derive a single situation as a result. To provide more suitable services, applications need to have a good understanding of what is actually happening in an environment. This understanding can include the reliability of this situation being recognised; and the implication on other situations given that this situation is recognised. For example, any situation mutually exclusive from this situation cannot happen, and any situation more general than this situation is happening. This implication on other situations can help a system to better configure applications.

Most existing research focuses on how to efficiently and accurately infer situations, but this should not be the final goal. A more interesting question would be: how do these recognised situations assist a system in providing users intuitive and less intrusive services. Compared to the body of work in sensor design and activity recognition techniques, the work in situation-aware applications is much less. Research needs to move towards realistic applications like [69], rather than toy or imaginary applications. With realistic applications, we can evaluate how much a user feels satisfied or disturbed by the services. The evaluation result might give us new insights from users' perspective, including (1) what situations are needed to be identified; for example, for home secure applications, grooming activities like styling hair may have much less importance than a cooking activity; (2) to what granularity an situation should be defined; for example, should we define an activity as fine-grained as stirring, chopping, or adding ingredients in a cooking activity.

### 7.3. Open Research Questions

In studying situations, researchers are more and more interested in recognising interleaved situations where more than one users are involved or more finer-grained temporally overlapping situations are involved. Because of the rich structure, hierarchical HMMs are the most popular technique that is used in identifying such complex situations [4]. However, the complexity of computation also increases greatly with the complexity of the structure.

Currently, an underlying assumption of situation identification is that situations are pre-defined in specification-based approaches or pre-labelled in supervised learning-based approaches. When it comes to short, non-repetitive, and unpredictable situations while significant to applications (like a heart attack), it would be difficult to spot them [156]. For continuous situations, it is still challenging in mining implicit and uncontinuous temporal sequences between them and detecting boundaries where situations change are.

As a bridge that links sensor data to applications, a situation model should not only be able to predict situations, but also to provide insights on how a system infers situations, and on how sensors perform, which is called *intelligibility* in [34, 93]. By analysing formed logical specifications' of situations, developers can learn which sensors are better in recognising a certain situation, how users behave or interact with sensors. The ability of users to understand system decision making in order to develop a mental model of the system is critical to its acceptance. Therefore, a challenge is to ensure that situation models are sufficiently informative and transparent to enable intelligibility.

Current research has largely focussed on data sets collected in research labs or by environment occupied by researchers. When real world environments are used, more complexities appear, such as situation interruption, multi-tasking, multiple users, unexpected user behaviour, as described by Logan *et al*, where they look at activity monitoring in a smart home. As part of this problem, the research community will need to examine new measures for evaluating reasoning techniques should be used. At present, the focus is on classification accuracies using traditional machine learning measures; that is obtaining the 'right' answer for as many instances, akin to classifying static knowledge such as documents. But in pervasive environments, situations are dynamic, of varied duration, sequential, interleaved; and application behaviours and transitions need to be smoothed and controlled. For example, rather than checking the proportion of a situation correctly recognised, it may be more useful to check whether an activity was detected at all over a period of time; e.g in a monitored smart home, did the user prepare breakfast today at all?. Boundaries between situations may be important for health applications [77]; for example, whether a person's heart rate has moved from normal to high within in a certain period of time. For the next phase of research, researchers should examine what real-world complexities need to be addressed, and what new measures should be considered for evaluations in the future.

One of the challenges in pervasive computing is the requirement to re-create a model of each new environment in which an application will reside. An activity monitoring application, for example, may need to cater for different sensors, different user behaviours and so forth when applied across different homes. With machine learning approaches, training data must be collected for any change in environment. This issue of 'transfer learning' [135] addresses the problem of how to use annotated training data from one environment to label training data from another environment. This is an important issue for machine-learning techniques in order to avoid the costly collection and annotation of data sets for each application environment.

## 8. Conclusion

In this paper we have described the state-of-the-art research in situation identification in pervasive computing. This research is challenged by the complexity of pervasive computing in terms of highly sensorised environments and contextual applications customised to a variety of factors. We have discussed different aspects of situation research: representation, specification, and reasoning, and have elicited the requirements and challenges in each aspect.

We have introduced the existing techniques in recognising situations, and compared them against the qualitative metrics. Based on the analysis, we suggest a hybrid approach of specification- and learning-based techniques, where a specification-based technique is responsible of knowledge representation and sharing while a learning-based technique is responsible of deriving new knowledge and dealing with uncertainty in sensor data. In the end we also discuss some open research opportunities in the area of situation identification.

## 9. Acknowledgements

This work is partially supported by the FP7 FET proactive project *SAPERE – self-aware pervasive service ecosystems*, under grant no. 25873. We thank all the reviewers for their constructive comments.

## References

[1] Abdelsalam, W., Ebrahim, Y., May 2004. A Bayesian-networks-based approach for managing uncertainty in location-tracking . In: Proceedings of the Canadian Conference on Electrical and Computer Engineering. Vol. 4. pp. 2205–2208.

[2] Agrawal, R., Srikant, R., 1995. Mining sequential patterns. In: Proceedings of International Conference on Data Engineering. Vol. 0. IEEE Computer Society, Los Alamitos, CA, USA, pp. 3–14.

[3] Anagnostopoulos, C. B., Ntarladimas, Y., Hadjiefthymiades, S., 2007. Situational computing: An innovative architecture with imprecise reasoning. Journal of System and Software 80 (12), 1993–2014.

[4] Atallah, L., Yang, G.-Z., 2009. The use of pervasive sensing for behaviour profiling  a survey. Pervasive and Mobile Computing, 447–464.

[5] Augusto, J., Nugent, C., Aug. 2004. The use of temporal reasoning and management of complex events in smart homes. In: ECAI' 04: Proceedings of European Conference on Artificial Intelligence. pp. 778–782.

[6] Augusto, J. C., Liu, J., McCullagh, P., Wang, H., Yang, J.-B., 2008. Management of uncertainty and spatio-temporal aspects for monitoring and diagnosis in a smart home. International Journal of Computational Intelligence Systems (4), 361–378.

[7] Aztiria, A., Augusto, J., Izaguirre, A., Cook, D., 2008. Learning accurate temporal relations from user actions in intelligent environments. In: Proceedings of the Symposium of Ubiquitous Computing and Ambient Intelligence.

[8] Aztiria, A., Cook, D., 2010. Automatic modeling of frequent user behaviours in intelligent environments. In: IE'10: proceedings of the International Conference on Intelligent Environments.

[9] Aztiria, A., Izaguirre, A., Augusto, J. C., 2010. Learning patterns in ambient intelligence environments: a survey. Artificial Intelligence Review 34 (1), 35–51.

[10] Bao, L., Intille, S. S., Apr. 2004. Activity recognition from user-annotated acceleration data. In: Pervasive' 04: Proceedings of the second International Conference on Pervasive Computing. Vienna, Austria, pp. 1–17.

[11] Baum, L. E., Petrie, T., Soules, G., Weiss, N., 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. The Annals of Mathematical Statistics 41 (1), 164–171.

[12] Baumgartner, N., Retschitzegger, W., Feb. 2007. Towards a situation awareness framework based on primitive relations. In: IDC '07: Proceedings of the international conference on Information, Decision and Control. pp. 291–295.

[13] Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., Riboni, D., June 2009. A survey of context modelling and reasoning techniques. Pervasive and Mobile Computing.

[14] Bharatula, N. B., Stäger, M., Lukowicz, P., Tröster, G., March 2005. Empirical Study of Design Choices in Multi-Sensor Context Recognition Systems. In: IFAWC: 2nd International Forum on Applied Wearable Computing. pp. 79–93.

[15] Bhatt, M., Loke, S., Jan. 2008. Modelling dynamic spatial systems in the situation calculus. Spatial Cognition and Computation 8, 86–130.

[16] Blanke, U., Schiele, B., 2009. Daily routine recognition through activity spotting. In: LoCA '09: Proceedings of the 4th International Symposium on Location and Context Awareness. Springer-Verlag, Berlin, Heidelberg, pp. 192–206.

[17] Brdiczka, O., Crowley, J. L., Reignier, P., 2009. Learning situation models in a smart home. IEEE Transactions on System, Man and Cybernetics, part B 39 (1).

[18] Brdiczka, O., Maisonnasse, J., Reignier, P., Crowley, J. L., 2009. Detecting small group activities from multimodal observations. International Journal of Applied Intelligence 30 (1), 47–57.

[19] Bui, H., Venkatesh, S., West, G., 2002. Policy recognition in the abstract hidden markov models. Journal of Artificial Intelligence Research 17, 451499.

[20] Bulling, A., Ward, J. A., Gellersen, H., Tröster, G., 2008. Robust recognition of reading activity in transit using wearable electrooculography. In: Pervasive '08: Proceedings of the 6th International Conference on Pervasive Computing. Springer-Verlag, Berlin, Heidelberg, pp. 19–37.

[21] Castellano, G., Villalba, S. D., Camurri, A., 2007. Recognising human emotions from body movement and gesture dynamics. In: ACII '07: Proceedings of the 2nd international conference on Affective Computing and Intelligent Interaction. Springer-Verlag, Berlin, Heidelberg, Lisbon, Portugal, pp. 71–82.

[22] Chang, K.-H., Hightower, J., Kveton, B., 2009. Inferring identity using accelerometers in television remote controls. In: Pervasive '09: Proceedings of the 7th International Conference on Pervasive Computing. Springer-Verlag, Berlin, Heidelberg, pp. 151–167.

[23] Chen, H., Finin, T., Joshi, A., May 2004. An Ontology for Context-Aware Pervasive Computing Environments. Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review 18 (3), 197–207.

[24] Chen, L., Nugent, C., Mulvenna, M., Finlay, D., Hong, X., 2009. Semantic smart homes: Towards knowledge rich assisted living environments. Intelligent Patient Management 189, 279–296.
[25] Cheung, K. S., Vogel, D., 2005. Complexity reduction in lattice-based information retrieval. Information Retrieval 8 (2), 285–299.
[26] Choujaa, D., Dulay, N., 2009. Activity inference through sequence alignment. In: LoCA '09: Proceedings of the 4th International Symposium on Location and Context Awareness. Springer-Verlag, Berlin, Heidelberg, pp. 19–36.
[27] Clarkson, B., Pentland, A., 1999. Unsupervised clustering of ambulatory audio and video. In: ICASSP '99: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing. pp. 3037–3040.
[28] Cohen, N. H., Lei, H., Castro, P., II, J. S. D., Purakayastha, A., 2002. Composing pervasive data using iQL. In: WMCSA '02: Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications. pp. 94–104.
[29] Cook, D. J., Augusto, J. C., Jakkula, V. R., August 2009. Ambient intelligence: Technologies, applications, and opportunities. Pervasive and Mobile Computing 5 (4), 277–298.
[30] Cook, D. J., Das, S. K., 2007. How smart are our environments? an updated look at the state of the art. Pervasive Mobile Computing 3 (2), 53–73.
[31] Costa, P. D., Guizzardi, G., Almeida, J. P. A., Pires, L. F., van Sinderen, M., 2006. Situations in conceptual modeling of context. In: EDOCW '06: Proceedings of the 10th IEEE on International Enterprise Distributed Object Computing Conference Workshops. IEEE Computer Society, Washington, DC, USA, pp. 6–16.
[32] Delir Haghighi, P., Krishnaswamy, S., Zaslavsky, A., Gaber, M. M., 2008. Reasoning about context in uncertain pervasive computing environments. In: EuroSSC '08: Proceedings of the 3rd European Conference on Smart Sensing and Context. Springer-Verlag, Berlin, Heidelberg, pp. 112–125.
[33] Dempster, A. P., 1968. A generalization of bayesian inference. the Royal Statistical Society, Series B 30, 205–247.
[34] Dey, A. K., 2009. Modeling and intelligibility in ambient environments. Ambient Intelligence and Smart Environments 1, 47–62.
[35] Dietterich, T. G., Ashenfelter, A., Bulatov, Y., 2004. Training conditional random fields via gradient tree boosting. In: ICML '04: Proceedings of the twenty-first international conference on Machine learning. ACM, Banff, Alberta, Canada, p. 28.
[36] Ding, Z., Peng, Y., 2004. A probabilistic extension to ontology language OWL. In: HICSS' 04: Proceedings of the 37th Annual Hawaii International Conference on System Sciences. pp. 1–10.
[37] Duong, T. V., Bui, H. H., Phung, D. Q., Venkatesh, S., 2005. Activity recognition and abnormality detection with the switching hidden semi-markov model. In: CVPR' 05: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition - Volume 1. San Diego, CA, USA, pp. 838–845.
[38] Fine, S., Singer, Y., Tishby, N., 1998. The hierarchical hidden markov model: Analysis and applications. Machine Learning 32, 41–62.
[39] Flury, B. N., 1984. Common principal components in k groups. Journal of the American Statistical Association 79, 892–898.
[40] Galton, A., 2004. Eventualities. MIT Press.
[41] Gottfried, B., Guesgen, H. W., Hübner, S., 2006. Spatiotemporal reasoning for smart homes. Designing Smart Homes 4008, 16–34.
[42] Gray, P. D., Salber, D., 2001. Modelling and using sensed context information in the design of interactive applications. In: EHCI '01: Proceedings of the 8th IFIP International Conference on Engineering for Human-Computer Interaction. Springer-Verlag, London, UK, pp. 317–336.
[43] Gu, T., Chen, S., Tao, X., Lu, J., 2010. An unsupervised approach to activity recognition and segmentation based on object-use fingerprints. Data and Knowledge Engineering 69 (6), 533–544.
[44] Gu, T., Pung, H. K., Zhang, D. Q., April 2004. A Bayesian approach for dealing with uncertain contexts. In: Proceedings of Advances in Pervasive Computing, coexisted with Pervasive' 04. Austrian Computer Society, pp. 205–210.
[45] Gu, T., Pung, H. K., Zhang, D. Q., 2005. A service-oriented middleware for building context-aware services. Journal of Network and Computer Applications 28 (1), 1–18.
[46] Gu, T., Wang, X. H., Pung, H. K., Zhang, D. Q., January 2004. An ontology-based context model in intelligent environments. In: CNDS' 04: Proceedings of the conference on Communication Network and Distributed Systems. pp. 270–275.
[47] Gu, T., Wu, Z., Tao, X., Pung, H. K., Lu, J., Mar. 2009. epsicar: An emerging patterns based approach to sequential, interleaved and concurrent activity recognition. In: PerCom'09: Proceedings of IEEE International Conference on Pervasive Computing and Communications. IEEE Computer Society, Los Alamitos, CA, USA, pp. 1–9.
[48] Gu, T., Wu, Z., Wang, L., Tao, X., Lu, J., July 2009. Mining emerging patterns for recognizing activities of multiple users in pervasive computing. In: MobiQuitous '09: Proceedings of the 6th Annual International on Mobile and Ubiquitous Systems: Networking Services. pp. 1 –2.
[49] Hamid, R., Maddi, S., Bobick, A., Essa, I., 2006. Unsupervised analysis of activity sequences using event-motifs. In: VSSN '06: Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks. ACM, New York, NY, USA, pp. 71–78.
[50] Hamid, R., Maddi, S., Johnson, A., Bobick, A., Essa, I., Isbell, C., 2009. A novel sequence representation for unsupervised analysis of human activities. Artificial Intelligence 173 (14), 1221–1244.
[51] Han, J., Kamber, M., 2006. Data mining:concepts and techniques. Morgan Kaufmann.
[52] Hannan, J., Bishop, J., 1997. A comparison of fast training algorithms over two real problems. In: Proceedings of Fifth International Conference on Artificial Neural Networks. pp. 1–6.
[53] Hasan, M. K., Rubaiyeat, H. A., Lee, Y.-K., Lee, S., 2008. A Reconfigurable HMM for Activity Recognition. In: ICACT'

08: Proceedings of 10th International Conference on Advanced Communication Technology. Vol. 1. pp. 843–846.
[54] Henricksen, K., Indulska, J., 2004. Modelling and using imperfect context information. In: Proceedings of PERCOM '04 Workshops. pp. 33 – 37.
[55] Henricksen, K., Indulska, J., Feb. 2006. Developing context-aware pervasive computing applications: models and approach. Pervasive and Mobile Computing 2 (1), 37–64.
[56] H.J.Zimmermann, 2000. An application-oriented view of modeling uncertainty. European Journal of Operational Research 122, 190–198.
[57] Hoffman, J. C., Murphy, R. R., Oct. 1993. Comparison of Bayesian and Dempster-Shafer theory for sensing: a practitioner's approach. In: Proceedings of Neural and Stochastic Methods in Image and Signal Processing II, Vol. 2032. pp. 266–279.
[58] Hong, X., Nugent, C., Mulvenna, M., McClean, S., Scotney, B., Devlin, S., 2009. Evidential fusion of sensor data for activity recognition in smart homes. Pervasive and Mobile Computing 5, 236–252.
[59] Hongeng, S., Nevatia, R., 2003. Large-scale event detection using semi-hidden markov models. In: ICCV' 03: Proceedings of the 9th IEEE International Conference on Computer Vision. pp. 1455–1462.
[60] Huynh, T., Fritz, M., Schiele, B., 2008. Discovery of activity patterns using topic models. In: UbiComp '08: Proceedings of the 10th international conference on Ubiquitous computing. ACM, New York, NY, USA, pp. 10–19.
[61] Ivanov, Y. A., Bobick, A. F., 2000. Recognition of visual activities and interactions by stochastic parsing. IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8), 852–872.
[62] Jakkula, V. R., Crandall, A. S., Cook, D. J., 2007. Knowledge discovery in entity based smart environment resident data using temporal relation based data mining. In: ICDMW '07: Proceedings of the Seventh IEEE International Conference on Data Mining Workshops. IEEE Computer Society, Washington, DC, USA, pp. 625–630.
[63] Jakkula, V. R., Crandall, A. S., Cook, D. J., 2009. Enhancing anomaly detection using temporal pattern discovery. Advanced Intelligent Enviroments, 175–194.
[64] Kalyan, A., Gopalan, S., V, S., 2005. Hybrid context model based on multilevel situation theory and ontology for contact centers. In: PERCOMW '05: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops. pp. 3–7.
[65] Kanda, T., Glas, D. F., Shiomi, M., Ishiguro, H., Hagita, N., 2008. Who will be the customer?: a social robot that anticipates people's behavior from their trajectories. In: UbiComp '08: Proceedings of the 10th international conference on Ubiquitous computing. ACM, Seoul, Korea, pp. 380–389.
[66] Karantonis, D. M., MR, N., M, M., NH, L., BG, C., 2006. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. IEEE transactions on information technology in biomedicine 10, 156–167.
[67] Klein, M., Schmidt, A., Lauer, R., 2007. Ontology-centred design of an ambient middleware for assisted living: The case of soprano. In: Kirste, T., Knig-Ries, B., Salomon, R. (Eds.), Proceedings of Towards Ambient Intelligence: Methods for Cooperating Ensembles in Ubiquitous Environments (AIM-CU), co exists with 30th Annual German Conference on Artificial Intelligence (KI 2007).
URL http://publications.andreas.schmidt.name/klein_schmidt_lauer_AIM-CU_KI07.pdf
[68] Krzanowski, W. J., 1979. Between-groups comparison of principal components. Journal of the American Statistical Association 74, 703–707.
[69] Kunze, K., Lukowicz, P., 2008. Dealing with sensor displacement in motion-based onbody activity recognition systems. In: UbiComp '08: Proceedings of the 10th international conference on Ubiquitous computing. ACM, New York, NY, USA, pp. 20–29.
[70] Latfi, F., Lefebvre, B., Descheneaux, C., 2007. Ontology-based management of the telehealth smart home, dedicated to elderly in loss of cognitive autonomy. In: Golbreich, C., Kalyanpur, A., Parsia, B. (Eds.), OWLED. Vol. 258 of CEUR Workshop Proceedings. CEUR-WS.org.
[71] Lei, H., Sow, D. M., John S. Davis, I., Banavar, G., Ebling, M. R., 2002. The design and applications of a context service. SIGMOBILE Mob. Computing Commun. Rev. 6 (4), 45–55.
[72] Levesque, H., Pirri, F., Reiter, R., 1998. Foundations for the situation calculus. Computer and Information Science 3.
[73] Li, B., Lijima, J., 2007. A survey on application of situation calculus in business information systems. In: ICCIT '07: Proceedings of the 2007 International Conference on Convergence Information Technology. IEEE Computer Society, Washington, DC, USA, pp. 425–424.
[74] Li, J., Liu, G., Wong, L., 2007. Mining statistically important equivalence classes and delta-discriminative emerging patterns. In: KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, New York, NY, USA, pp. 430–439.
[75] Liao, L., Fox, D., Kautz, H., 2007. Extracting places and activities from gps traces using hierarchical conditional random fields. International Journal on Robotics Research 26 (1), 119–134.
[76] Liu, D. C., Nocedal, J., 1989. On the limited memory BFGS method for large scale optimization. mathematical Programming 45, 503–528.
[77] Logan, B., Healey, J., Philipose, M., Tapia, E. M., Intille, S. S., September 2007. A long-term evaluation of sensing modalities for activity recognition. In: UbiComp'07: Proceedings of the 9th international conference on Ubiquitous computing. Springer-Verlag, Innsbruck, Austria, pp. 483–500.
[78] Loke, S. W., Sep. 2004. Logic programming for context-aware pervasive computing: Language support, characterizing situations, and integration with the web. In: WI' 04: Proceedings of 2004 IEEE/WIC/ACM International Conference on Web Intelligence. Beijing, China, pp. 44–50.
[79] Loke, S. W., 2004. Representing and reasoning with situations for context-aware pervasive computing: a logic program-

ming perspective. Knowledge Engineering Review 19 (3), 213–233.
[80] Loke, S. W., 2010. Incremental awareness and compositionality: A design philosophy for context-aware pervasive systems. Pervasive Mob. Comput. 6 (2), 239–253.
[81] Lowd, D., Domingos, P., 2005. Naive bayes models for probability estimation. In: ICML '05: Proceedings of the 22nd international conference on Machine learning. ACM, Bonn, Germany, pp. 529–536.
[82] Lühr, S., West, G., Venkatesh, S., 2007. Recognition of emergent human behaviour in a smart home: A data mining approach. Pervasive and Mobile Computing 3 (2), 95–116.
[83] Luo, H., Fan, J., 2006. Building concept ontology for medical video annotation. In: MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia. ACM, New York, NY, USA, pp. 57–60.
[84] Matheus, C. J., Kokar, M. M., Baclawski, K., Jul. 2003. A core ontology for situation awareness. In: Proceedings of the Sixth International Conference of Information Fusion. Versatile Information Systems, Inc., Queensland, Australia, pp. 545–552.
[85] Mathie, M. J., Celler, B. G., Lovell, N. H., Coster, A. C. F., 2006. Classification of basic daily movements using a triaxial accelerometer. Classification of basic daily movements using a triaxial accelerometer 42, 679–687.
[86] Maurer, U., Smailagic, A., Siewiorek, D. P., Deisher, M., 2006. Activity recognition and monitoring using multiple sensors on different body positions. In: BSN '06: Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks. pp. 113–116.
[87] McCarthy, J., Apr. 2000. Situation calculus with concurrent events and narrative. http://www-formal.stanford.edu/jmc/narrative/narrative.html, web only.
[88] McCarthy, J., Buvac, S., 1994. Formalizing context (expanded notes). Tech. rep., Stanford University, Stanford, CA, USA.
[89] McCowan, I., Gatica-Perez, D., Bengio, S., Lathoud, G., Barnard, M., Zhang, D., 2005. Automatic analysis of multimodal group actions in meetings. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 305–317.
[90] Mckeever, S., Ye, J., Coyle, L., Bleakley, C., Dobson, S., 2010. Activity recognition using temporal evidence theory. Journal of Ambient Intelligence and Smart Environment 2 (3), 253–269.
[91] McKeever, S., Ye, J., Coyle, L., Dobson, S., May 2008. A multilayered uncertainty model for context aware systems. In: Adjunct Proceedings of the international conference on Pervasive Computing: Late Breaking Result. Sydney, Australia, pp. 1–4.
[92] McKeever, S., Ye, J., Coyle, L., Dobson, S., 2009. A context quality model to support transparent reasoning with uncertain context. In: Rothermel, K., Fritsch, D., Blochinger, W., Drr, F. (Eds.), QuaCon' 09: Proceedings of the 1st International Workshop on Quality of Context. Vol. 5786 of Lecture Notes in Computer Science. Springer, pp. 65–75. URL http://dblp.uni-trier.de/db/conf/quacon/quacon2009.html#McKeeverYCD09
[93] McKeever, S., Ye, J., Coyle, L., Dobson, S., 2009. Using dempster-shafer theory of evidence for situation inference. In: Barnaghi, P. M., Moessner, K., Presser, M., Meissner, S. (Eds.), EuroSSC 2009: Proceedings of the 4th European Conference on Smart Sensing and Context. Vol. 5741 of Lecture Notes in Computer Science. Springer, pp. 149–162.
[94] Michael, R.and Funder, D., 2004. Situational similarity and behavioral consistency: Subjective, objective, variable-centered, and person-centered approaches. Journal of Research in Personality 38, 421447.
[95] Minnen, D., Starner, T., Ward, J., Lukowicz, P., Troster, G., July 2005. Recognizing and discovering human actions from on-body sensor data. In: ICME' 05: Proceedings of the International Conference on Multi-media and Expo. Amsterdam, Denmark, pp. 1545–1548.
[96] Moore, D., Essa, I., 2002. Recognizing multitasked activities from video using stochastic context-free grammar. In: Eighteenth national conference on Artificial intelligence. American Association for Artificial Intelligence, Menlo Park, CA, USA, pp. 770–776.
[97] Mrohs, B., Luther, M., Vaidya, R., Wagner, M., Steglich, S., Kellerer, W., Arbanowski, S., 2005. Owl-sf-a distributed semantic service framework. In: Proceedings of the 1st Workshop on Context Awareness for Proactive Systems. pp. 66–78.
[98] Mühlenbrock, M., Brdiczka, O., Snowdon, D., Meunier, J.-L., Mar. 2004. Learning to detect user activity and availability from a variety of sensor data. In: PERCOM '04: Proceedings of the Second IEEE International Conference on Pervasive Computing and Communications. IEEE Computer Society, Orlando, Florida, p. 13.
[99] neural networks, A., 2007. Neural networks: A requirement for intelligent systems. http://www.learnartificialneuralnetworks.com/.
[100] Nevatia, R., Hobbs, J., Bolles, B., 2004. An ontology for video event representation. In: CVPRW '04: Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop Volume 7. IEEE Computer Society, Washington, DC, USA, p. 119.
[101] Nguyen, N. T., Phung, D. Q., Venkatesh, S., Bui, H., 2005. Learning and detecting activities from movement trajectories using the hierarchical hidden markov models. In: CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2. IEEE Computer Society, Washington, DC, USA, pp. 955–960.
[102] Oliver, P., Monk, A., Xu, G., Hoey, J., 2009. Ambient Kkitchen: designing situated services using a high fidelity prototyping environment. In: Proceedings of the Workshop on Affect and Behaviour-Related Assistance in the Support of the Elderly (PETRA'09).
[103] Palmes, P., Pung, H. K., Gu, T., Xue, W., Chen, S., 2010. Object relevance weight pattern mining for activity recognition and segmentation. Pervasive Mobile Computing 6 (1), 43–57.
[104] Patel, S. N., Robertson, T., Kientz, J. A., Reynolds, M. S., Abowd, G. D., 2007. At the flick of a switch: detecting and classifying unique electrical events on the residential power line. In: UbiComp'07: Proceedings of the 9th international conference on Ubiquitous computing. Springer-Verlag, Berlin, Heidelberg, pp. 271–288.

[105] Patterson, D. J., Fox, D., Kautz, H., Philipose, M., 2005. Fine-grained activity recognition by aggregating abstract object usage. In: ISWC '05: Proceedings of the Ninth IEEE International Symposium on Wearable Computers. pp. 44–51.
[106] Patterson, D. J., Liao, L., Fox, D., Kautz, H., 2003. Inferring high-level behavior from low-level sensors. In: Ubicomp' 03: Proceedings of Proceedings of the 5th international conference on Ubiquitous computing. Springer Berlin/Heidelberg, pp. 73–89.
[107] Pearlmutter, B., 1995. Gradient calculations for dynamic recurrent neural networks: a survey. IEEE Transaction on Neural Networks 6, 1212–1228.
[108] Pedrycz, W., Camargo, H., December 2003. Fuzzy timed petri nets. Fuzzy Sets and Systems 140 (2), 301–330.
[109] Perkowitz, M., Philipose, M., Fishkin, K., Patterson, D. J., 2004. Mining models of human activities from the web. In: WWW '04: Proceedings of the 13th international conference on World Wide Web. ACM, New York, NY, USA, pp. 573–582.
[110] Rabiner, L. R., 1990. A tutorial on hidden markov models and selected applications in speech recognition. Readings in Speech Recognition, 267–296.
[111] Ranganathan, A., Al-Muhtadi, J., Campbell, R. H., 2004. Reasoning about uncertain contexts in pervasive computing environments. IEEE Pervasive Computing 03 (2), 62–70.
[112] Ranganathan, A., Al-Muhtadi, J., Chetan, S., Campbell, R., Mickunas, M. D., 2004. Middlewhere: a middleware for location awareness in ubiquitous computing applications. In: Proceedings of Middleware '04. New York, USA, pp. 397–416.
[113] Ranganathan, A., Mcgrath, R. E., Campbell, R. H., Mickunas, M. D., 2004. Use of ontologies in a pervasive computing environment. Knowledge Engineering Review 18 (3), 209–220.
[114] Reignier, P., Brdiczka, O., Vaufreydaz, D., Crowley, J. L., Maisonnasse, J., 2007. Context-aware environments: from specification to implementation. Expert Systems 24 (5), 305–320.
[115] Riboni, D., Bettini, C., 2009. Context-aware activity recognition through a combination of ontological and statistical reasoning. In: Proceedings of the international conference on Ubiquitous Intelligence and Computing. Springer Berlin/Heidelberg, pp. 39–53.
[116] Riedmiller, M., Braun, H., 1993. A direct adaptive method for faster backpropagation learning: The rprop algorithm. In: Proceedings of the IEEE International Conference on Neural Networks. pp. 586–591.
[117] Ryoo, M. S., Aggarwal, J. K., 2006. Recognition of composite human activities through context-free grammar based representation. In: CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, Washington, DC, USA, pp. 1709–1718.
[118] SanMiguel, J. C., Martinez, J. M., Garcia, A., 2009. An ontology for event detection and its application in surveillance video. In: Proceedings of SIXTH IEEE International Conference on Advanced Video and Signal Based Surveillance. Vol. 0. IEEE Computer Society, Los Alamitos, CA, USA, pp. 220–225.
[119] Schmidt, A., Beigl, M., Gellersen, H.-W., 1999. There is more to context than location. Computers and Graphics 23 (6), 893–901.
[120] Schmidtke, H. R., Woo, W., 2009. Towards ontology-based formal verification methods for context aware systems. In: Pervasive '09: Proceedings of the 7th International Conference on Pervasive Computing. Springer-Verlag, Berlin, Heidelberg, pp. 309–326.
[121] Sentz, K., Ferson, S., 2002. Combination of evidence in dempster-shafer theory. Tech. Rep. SAND 2002-0835, Sandia National Laboratories.
[122] Shafer, G., 1976. A Mathematical Theory of Evidence. Princeton University Press.
[123] Shannan, C. E., Weaver, W., Oct. 1948. The mathematical theory of communication. Bell System Technical Journal, 27, 379–423, 623–656.
[124] Smirnov, A., Kashevnik, A., Levashova, T., Pashkin, M., Shilov, N., May 2007. Situation modeling in decision support systems. In: KIMAS' 07: Proceedings of International Conference on Integration of Knowledge Intensive Multi-Agent Systems. pp. 34–39.
[125] Stevenson, G., Knox, S., Dobson, S., Nixon, P., June 2009. Ontonym: A Collection of Upper Ontologies for Developing Pervasive Systems. In: Proceedings of the Workshop on Context, Information And Ontologies. Heraklion, Greece.
[126] Stevenson, G., Ye, J., Dobson, S., Nixon, P., 2009. Loc8: A location model and extensible framework for programming with location. IEEE Pervasive Computing 9, 28–37.
[127] Strang, T., Linnhoff-Popien, C., Sepember 2004. A context modeling survey. In: Proceedings of the Workshop on Advanced Context Modelling, Reasoning and Management as part of UbiComp 2004. Nottingham/England.
[128] Tapia, E. M., Choudhury, T., Philipose, M., May 2006. Building reliable activity models using hierarchical shrinkage and mined ontology. In: Pervasive' 06: Proceedings of the 4th International Conference on Pervasive Computing. Vol. 3968 of Lecture Notes in Computer Science. Springer, Dublin, Ireland, pp. 17–32.
[129] Tapia, E. M., Intille, S. S., Larson, K., 2004. Activity recognition in the home using simple and ubiquitous sensors. In: Pervasive' 04: Proceedings of the international conference on Pervasive Computing. pp. 158–175.
[130] Thomson, G., Terzis, S., Nixon, P., 2006. Situation determination with reusable situation specifications. In: Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops. pp. 620–623.
[131] Truong, B. A., Lee, Y.-K., Lee, S.-Y., 2005. Modeling uncertainty in context-aware computing. In: ICIS '05: Proceedings of the Fourth Annual ACIS International Conference on Computer and Information Science. pp. 676–681.
[132] Turaga, P., Chellappa, R., Subrahmanian, V. S., Udrea, O., Nov. 2008. Machine recognition of human activities: A survey. IEEE Transactions on Circuits and Systems for Video Technology 18, 1473–1488.
[133] Vail, D. L., Veloso, M. M., 2008. Feature selection for activity recognition in multi-robot domains. In: AAAI'08: Pro-

ceedings of the 23rd national conference on Artificial intelligence. AAAI Press, Chicago, Illinois, pp. 1415–1420.
[134] Vail, D. L., Veloso, M. M., Lafferty, J. D., 2007. Conditional random fields for activity recognition. In: AAMAS '07: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems. ACM, New York, NY, USA, pp. 1–8.
[135] van Kasteren, T., Englebienne, G., Kröse, B., May 2010. Transferring knowledge of activity recognition across sensor networks. In: Pervasive '10: Proceedings of the 8th International Conference on Pervasive Computing. Springer-Verlag, Helsinki, Finland, pp. 283–300.
[136] van Kasteren, T., Krose, B., Sep. 2007. Bayesian activity recognition in residence for elders. In: IE' 07: Proceedings of the third IET International Conference on Intelligent Environments. pp. 209–212.
[137] van Kasteren, T., Noulas, A., Englebienne, G., Kröse, B., Sep. 2008. Accurate activity recognition in a home setting. In: UbiComp '08: Proceedings of the 10th International Conference on Ubiquitous Computing. ACM, Seoul, Korea, pp. 1–9.
[138] Wang, L., Gu, T., Chen, H., Tao, X., Lu, J., 2010. Real-time activity recognition in wireless body sensor networks: From simple gestures to complex activities. In: Proceedings of the International Workshop on Real-Time Computing Systems and Applications. Vol. 0. IEEE Computer Society, Los Alamitos, CA, USA, pp. 43–52.
[139] Wang, L., Gu, T., Tao, X., Lu, J., 2009. Sensor-based human activity recognition in a multi-user scenario. In: AmI '09: Proceedings of the European Conference on Ambient Intelligence. Springer-Verlag Berlin Heidelberg, pp. 78–87.
[140] Weißsenberg, N., Gartmann, R., Voisard, A., 2006. An ontology-based approach to personalized situation-aware mobile service supply. Geoinformatica 10 (1), 55–90.
[141] Welch, L. R., December 2003. Hidden markov models and the baum-welch algorithm. IEEE Information Theory Society Newsletter 53 (4).
[142] Witten, I. H., Frank, E., 2005. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann.
[143] Wojek, C., Nickel, K., Stiefelhagen, R., Sep. 2006. Activity recognition and room-level tracking in an office environment. In: MFI '06: Proceedings of 2006 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems. pp. 25–30.
[144] Yamada, N., Sakamoto, K., Kunito, G., Isoda, Y., Yamazaki, K., Tanaka, S., 2007. Applying ontology and probabilistic model to human activity recognition from surrounding things. IPSJ Digital Courier 3, 506–517.
[145] Yang, J.-Y., Wang, J.-S., Chen, Y.-P., 2008. Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers. Pattern Recognition Letter 29 (16), 2213–2220.
[146] Yau, S. S., Liu, J., 2006. Hierarchical situation modeling and reasoning for pervasive computing. In: SEUS-WCCIA '06: Proceedings of the The Fourth IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems, and the Second International Workshop on Collaborative Computing, Integration, and Assurance. IEEE Computer Society, Washington, DC, USA, pp. 5–10.
[147] Ye, J., Clear, A., Coyle, L., Dobson, S., August 2009. On using temporal semantics to create more accurate human-activity classifiers. In: Proceedings of 20th Irish Conference on Artificial Intelligence and Cognitive Science. Dublin, Ireland, pp. 274–283.
[148] Ye, J., Coyle, L., Dobson, S., Nixon, P., December 2007. Ontology-based models in pervasive computing systems. The Knowledge Engineering Review 22, 315–347.
[149] Ye, J., Coyle, L., Dobson, S., Nixon, P., August 2007. Using situation lattices to model and reason about context. In: MRC 2007: Proceedings of the Workshop on modeling and reasoning on context (coexist with CONTEXT'07). Roskilde, Denmark, pp. 1–12.
[150] Ye, J., Coyle, L., Dobson, S., Nixon, P., Mar. 2009. Using situation lattices in sensor analysis. In: PERCOM' 09: Proceedings of the 2009 IEEE International Conference on Pervasive Computing and Communications. Texas, USA, pp. 1–11.
[151] Ye, J., Dobson, S., 2010. Exploring semantics in activity recognition using context lattices. Journal of Ambient Intelligence and Smart Environments 2 (4), 389–407.
[152] Ye, J., McKeever, S., Coyle, L., Neely, S., Dobson, S., July 2008. Resolving uncertainty in context integration and abstraction. In: ICPS' 08: Proceedings of the international conference on Pervasive Services. ACM, Sorrento, Italy, pp. 131–140.
[153] Yoon, H., Yang, K., Shahabi, C., 2005. Feature subset selection and feature ranking for multivariate time series. IEEE Trans. on Knowl. and Data Eng. 17 (9), 1186–1198.
[154] Zadeh, L. A., 1965. Fuzzy sets. Information and Control 8, 338–353.
[155] Zhang, D., Cao, J., Zhou, J., Guo, M., 2009. Extended dempster-shafer theory in context reasoning for ubiquitous computing environments. In: CSE '09: Proceedings of the 2009 International Conference on Computational Science and Engineering. IEEE Computer Society, Washington, DC, USA, pp. 205–212.
[156] Zinnen, A., Wojek, C., Schiele, B., 2009. Multi activity recognition based on bodymodel-derived primitives. In: LoCA '09: Proceedings of the 4th International Symposium on Location and Context Awareness. Springer-Verlag, Berlin, Heidelberg, pp. 1–18.