J Inf Process Syst, Vol.10, No.3, pp.335~354, September 2014
http://dx.doi.org/10.3745/JIPS.04.0005

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

Training-Free Fuzzy Logic Based Human Activity Recognition

Eunju Kim* and Sumi Helal*

Abstract—The accuracy of training-based activity recognition depends on the training procedure and the extent to which the training dataset comprehensively represents the activity and its varieties. Additionally, training incurs substantial cost and effort in the process of collecting training data. To address these limitations, we have developed a training-free activity recognition approach based on a fuzzy logic algorithm that utilizes a generic activity model and an associated activity semantic knowledge. The approach is validated through experimentation with real activity datasets. Results show that the fuzzy logic based algorithms exhibit comparable or better accuracy than other training-based approaches.

Keywords—Activity Semantic Knowledge, Fuzzy Logic, Human Activity Recognition, Multi-Layer Neural Network

1. Introduction

Activity recognition can significantly empower many human centric applications in a variety of domains, such as healthcare, elder care, and other quality of life concerns [1,2]. For instance, in telehealth systems, caregivers are empowered by activity recognition instead of relying on and sifting through large datasets for analysis–a process hat is highly error prone. Human activity recognition research is surveyed in [3-5]. The applicability of activity recognition in real world applications is limited however by the low recognition accuracy—which is often unacceptable. For example, taking medications regularly is very important for patients. A healthcare system that does not recognize 'taking a medicine' activity accurately may produce incorrect recommendations. Such errors may lead to inadequate dosage or over-dosage. Significant activity recognition research is focused on finding algorithms with improved accuracy [3-5]. However, accuracy of current state of the art activity recognition technology remains inadequate leaving room for significant improvements before activity recognition can go mainstream in application developments.

1.1 Challenges of Human Activity Recognition

Human activity recognition is challenging because of the following characteristics of human activities.

1) Ambiguity

Recognizing human activities can produce ambiguous results because of several reasons. First,

※ This research was supported by a United States National Institutes of Health Grant (No. 5R21DA024294).
Manuscript received July 09, 2014; accepted August 01, 2014.
Corresponding Author: Sumi Helal (helal@cise.ufl.edu)
* Mobile and Pervasive Computing Lab., Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL, USA ({helal, ejkim}@cise.ufl.edu)

sometimes, people initiate an activity but do not complete it. These partially performed activities result in recognition ambiguity. For example, people may ingest a small amount of food but not complete their meal. Second when an activity initiates, there may not be enough information to determine the activity with a high confidence level. It takes a finite time interval to determine the performed activity.

2) Variety

Due to the variety of human activities, it is difficult to recognize all activities. An activity may be performed in many different ways. For instance, even though 'MakingHotTea' is a simple activity, there are several ways to make a hot tea. Some people boil water using a stove. Other people may use microwave.

1.2 Limitations of Existing Training Based Activity Recognition Technologies

1) Accuracy

Many activity recognition techniques are based on supervised machine learning algorithms that require training. In such technologies, activity recognition accuracy is highly dependent upon the training data and training process. Due to the variety of human activities, it is almost impossible to collect a comprehensive dataset for training a system to achieve high accuracy. Therefore, it is important to find a new activity recognition approach that can show consistent accuracy even without training.

2) Programmability

In activity recognition, programmability is the capability to support activity recognition system design change according to new application requirements or activity recognition environment changes. Many training based algorithms do not offer adequate programmability because new training is required whenever there is a change in activity recognition system. These repeated training is prohibitive due to cost and resource constraints.

1.3 Motivation

To address the issues of accuracy and programmability, we have developed a new activity recognition approach that does not require training. Our ultimate goal is to develop a system whose activity recognition performance is comparable to the activity recognition performance of humans. Hence, before developing an activity recognition approach, we compared machine based activity recognition methods with activity recognition methods used by humans. We found three major differences, which are enumerated below.

First, humans have very accurate activity model. The source of this high accuracy is their knowledge. To illustrate, when people think about an activity, they are able to estimate when and where it can be performed, how to perform it, and why it should be performed. Also, the activity model of a human keeps evolving as the human gains more knowledge. Therefore, in addition to building an activity model with activity knowledge, the activity model should be amenable to refinement and expansion.

Second, humans utilize activity semantic knowledge to determine a performed activity through logical inferences. Activity semantic knowledge is the knowledge that is not directly related to performing a specific activity, yet it provides for significant information that helps determine if an activity actually happened. For example, if cooking and sleeping activities of a person are recognized at the same time, it is implied that at least one of the recognition results is not correct because people cannot cook while sleeping. People utilize this semantic knowledge to recognize an activity accurately. Hence, if an activity recognition system utilizes activity semantic knowledge, it can enhance accuracy.

Third, people can recognize ambiguous activities well and describe the activity precisely using a variety of linguistic words. In other words, when an activity is ambiguously performed, people are able to describe the activity linguistically. They do not always make a clear recognition decision. For instance, when a user takes one spoonful of food, the user probably did not have a complete meal. In this case, people describe the patient's activity as "the patient had a small amount of food" or "the patient had one spoonful only." The linguistic description is more accurate than a recognition system that generates a binary output, i.e., "the patient had food" or "the patient didn't have food." Activity recognition system should be able to use linguistic terms to avoid recognition errors of ambiguously performed activities.

To develop an activity recognition system that can support all requirements for mimicking humans' activity recognition as described above, we developed a generic activity modeling framework and a set of activity semantics that it can utilize. Based on our new modeling framework—a hierarchical framework for modeling compositional relationship between an activity and components of the activity, we developed a fuzzy logic based activity recognition algorithm. Our approach realizes three key advantages.

1) No training required

Our approach doesn't require training. An existing activity recognition system may need to evolve because of several reasons such as changes in the target activities or sensor replacement due to advances in technology. Whenever, there is a change, training based activity recognition systems need a new set of training data and the activity recognition system must be retrained to reflect the change. Because our activity recognition system always cooperate with the activity models and activity semantics through direct querying instead of training, activity model change is carried over to the activity recognition system almost immediately without incurring too much additional training cost.

2) Model-recognizer synergy

Second, fuzzy logic is synergistic with our activity semantics because some semantics have characteristic of fuzziness. To illustrate, some activities, such as 'watching TV,' can be performed concurrently with many other activities. 'Sleeping' cannot usually be performed concurrently with other activities. And some activities, such as 'eating' or 'talking' are partially concurrent. This difference of activity concurrency can be represented using a fuzzy value. Additionally, fuzzy logic has fuzzy operations and inference capability that can handle this semantic knowledge. For example, if 'sleeping' and 'cooking,' which can never be concurrent, are recognized at the same time, fuzzy inference can be used to flag recognition inaccuracy.

3) Higher precision

Third, fuzzy logic based activity recognition approach can represent a recognized activity more precisely using linguistic output. Because linguistics is one of the main features of fuzzy logic theory, fuzzy logic can represent this linguistic terms easily via fuzzy membership function.

The rest of this paper is organized as follows. In the next sections, we review several related works. We then present our new activity modeling technique based on the generic activity framework (GAF) and associated activity semantics. We then describe our proposed fuzzy logic based activity recognition algorithm along with its implementation. Finally, experiments and results for validation of our approach are presented.

2. LITERATURE REVIEW

2.1 Review of Activity Framework–Activity Theory

L. S. Vygotsk who was a psychologist during 1920s and 1930s found activity theory. Later, A. N. Leontjev and A. R. Lurija further developed the activity theory [3,6-8]. Activity theory was first applied to human-computer interaction (HCI) in the early 1980s [7]. Recently, it is applied implicitly or explicitly in a lot of activity recognition model. The activity theory has four components such as subject, tool, objective, and outcome as shown in Fig. 1.

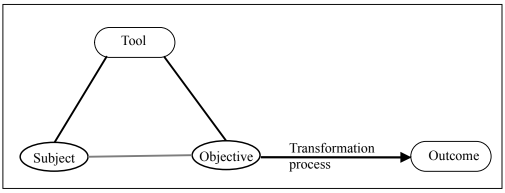

Fig. 1. Structure of activity theory.

A subject is a person who performs an activity. An objective is a plan or common idea that can be shared for manipulation and transformation by the participants of the activity. Tool is an artifact a subject uses to fulfill an objective. Outcome is another artifact or activity that are result of the activity. Transforming the objective into an outcome motivates the performing of an activity. For instance, having one's own house is an objective and the purchased house is the outcome. Transforming an object into an outcome requires various tools including documents, equipment, devices, etc. These relationships among components are presented with lines in Fig. 1.

Gray line between subject and objective represents a mediated relationship whereas bold line indicates direct relationship between components. Subject and tool have a direct relationship because a subject uses a tool in person. A tool mediates the relationship between an objective and subject, which is represented in gray line, because subjects achieve their objective using tools. Even though activity theory is well known and is often used in activity recognition research, it has some limitations. First, tool and object are not distinguish in activity theory even though it is necessary to distinguish them because the same artifact item may be used as tool in an activity, but object in other activities. In this case, the item has different meaning for each activity. For example, when a dish is used as a tool for cooking, it implies it contains food. On the other hand, if it is an object for washing dish activity, it means that it is an empty dish. Second, it is difficult to represent a temporal relationship between activities in activity theory due to the fact that activity theory focuses on the relationship between components such as subject, activity objective, tool and outcome rather than the relationship between activities.

2.2 Review of Probabilistic Activity Recognition Algorithm

In probabilistic activity recognition, it is assumed that human activities are continuously performed and each activity is a sequential composition of activity components, such as motions, operations or actions according to a temporal sequence [5]. According to this idea, several probabilistic models including hidden Markov model (HMM) or the conditional random field (CRF) model have been used to build an activity model because they are suitable for handling temporal sensor data.

2.2.1 HMM

HMM is a probabilistic function of Markov chains based on the first order Markov assumption of transition [9]. The basic idea of Markov chain of order m is that the future state depends on the past m numbers of states. Therefore, for HMM that is based on the first order Markov assumption, the future state depends only on the current state, not on past states [9]. Also HMM is a model that is used for generating hidden states from observable data. HMM determines the hidden state sequence (y1, y2, ..., yt) that corresponds to the observed sequence (x1, x2, ..., xt) [5]. In activity recognition, hidden state is human activities and therefore, HMM recognizes activities from both sensor observation and recognized activities in previous time according to the first order Markov chain. Moreover, HMM is also a generative and directed graph model [5,9]. Generative model means that observation data is randomly generated. In other words, it should enumerate all possible random cases in the model. Directed graph is used to capture orders between states. Therefore, a generative and directed graph model in activity recognition implies it should find all possible sequences of sensor observations of an activity.

However, many activities may have non-deterministic natures in practice, where some steps of the activities may be performed in any order. In practice, because many activities are concurrent or interleaved with other activities, HMM has difficulty in representing multiple interacting activities (concurrent or interleaved) [5,9]. Also HMM is incapable of capturing long-range or transitive dependencies of the observations due to its very strict independence assumptions on the observations. Therefore, enumerating all possible observation cases and orders is difficult for a practical activity recognition system. Furthermore, missing an observation or an order will cause the HMM to produce errors in the model.

2.2.2 CRF

CRF is a more flexible alternative to the HMM, which relaxes the strict assumption of HMM [4,5,10,11]. In other words, CRF solves the issues of HMM by neglecting the order constraint. Like HMM, CRF is also used to determine a hidden state transition from randomly generated observation sequences. However, CRF is a discriminative model, which does not generate all possible cases from the joint distribution of x and y. Therefore, CRF does not include arbitrarily complicated features of the observed variables into the model. Also, CRF is an undirected acyclic graph, flexibly capturing any relation between an observation variable and a hidden state [4,5]. Because CRF does not consider order, it considers only relationships, such as state feature function (relationship between observations over a period of time and activities) and transition feature function (relationship between past activities and future activities). Even though CRF removes order constraint from an activity model, CRF could outperform HMM [4].

3. GAF and Associated Activity Semantic Knowledge

3.1 GAF

We developed a GAF that has a hierarchical structure in which each layer consists of activity components. In total, there are eight components in the GAF. We chose the eight components according to 5W1H framework and we also found the eight is the most influential. In addition to the framework, we also add context because it is important to understand activities. The detailed description of the eight components is given below:

- Subject - is the actor of an activity.
- Time - when an activity is performed consists of start time and end time.
- Location - is the place where an activity is performed.
- Motive - is the reason why a subject performs a specific activity. Motive is the objective in activity theory as shown in Fig. 1. However, detecting motive is challenging. To determine motive, some artificial intelligent reasoning technique may be required.
- Tool - is an artifact that a subject uses to perform an activity. Tool provides essential information to classify activities. For example, a spoon or a fork is a tool for eating or cooking.
- Object - can also be any artifact like tool. However, object is the target of an activity whereas a subject uses a tool. Distinction between tool and object is important for accurate activity recognition because some artifacts are both tool and object depending on an activity.
- Context - is information, which is used to determine the situation where an activity is performed.

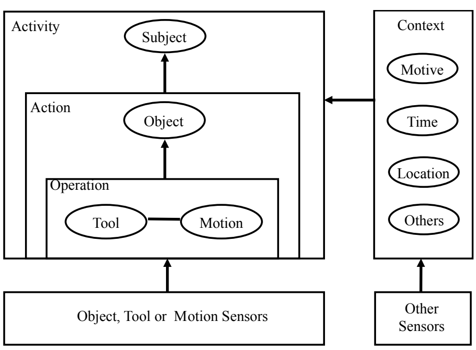

Fig. 2. Hierarchical structure of a generic activity framework.

Fig. 2 shows a composition diagram of the GAF. Rectangles are layers and ellipses represent components. The detailed description for each layer is given below:

- Sensors are installed in the pervasive space (e.g., a smart home) to collect event information of the space. Based on the source of sensor data, sensor is classified into four types: motion, tool, object, and context sensor.
- Operation is a composition of tool and motion. The user operates tools with specific motion. For example, if computer is a tool, some hand or finger motion will be performed for typing a keyboard.
- Action is determined by combination of operation and object. For example, if a user types a command to open a file, typing on the keyboard is an operation and the file is an object and

this combination is open file action.
- Activity is a collection of actions. Activity may involve multiple actions according to procedural steps.

3.2 Activity Semantics-Enriched GAF

Even though a GAF describes the composition hierarchy of activity components, it does not contain detailed activity semantic knowledge, such as role of an activity component, or relationships with other components. The activity semantics should be represented in an activity model because they are important for classifying an activity. For example, eating is composed of three actions, such as picking food, chewing food, and swallowing food. In this case, if only picking food is detected, then it is not clear whether eating is really performed or not because we are not sure the person completes the activity through chewing and swallowing the food or not. Activity semantics reduce these kinds of ambiguity. We introduce five activity semantics: dominance semantics, mutuality semantics, order semantics, activity effect, and activity life cycle (ALC).

1) Dominance semantics

In the hierarchical composition structure of an activity model, parent component may have several children components. Even though they are children components of same parent, the contribution of children may be different. Some children components are dominantly essential component of the activity whereas some are not. According to the dominance, we classify them as key or optional components.
- Key component is a mandatory component for identifying an activity. If an activity has multiple key components, all of them are required to agree with the activity. Otherwise, the activity is not considered performed.
- Optional component is not a key component of the activity. It is possible to omit an optional component because it does not always affect activity classification.

2) Mutuality semantics

This semantic knowledge is used to determine whether multiple activities can be concurrently performed or not.
- Concurrent components can be performed with other components together.
- Exclusive component cannot be performed simultaneously with another components. For example, sleeping is an exclusive activity because people cannot usually perform other activities when they sleep.
- Ordinary components are partially exclusive and concurrent. For example, when people sit, they cannot run at the same time. In this case, they are exclusive. But they can sit and sing a song concurrently. Hence, 'sit' is both partially exclusive and concurrent activity.

3) Order semantics

Some activities like an instruction should follow a procedural sequence. However, many activities have flexible order or do not have any order. Therefore, the role of order among activity components should be considered depending on the activity.
- Strong order requires that activity components (e.g., actions) should be always performed in a

specific order. For instance, waking-up definitely comes after sleeping.

- Weak order means that for many activities, their action components are performed according to a flexible order.
- Skip chain order has a temporal gap between two activities when an activity is interrupted by other activities. To illustrate, eating is usually performed immediately after cooking, but sometimes we can do other activities between them.

4) Effect semantics

Effect is caused by an activity. For example, the eating activity has effects, such as 'increasing glucose level' or 'increasing body temperature.' Activity effect is used to verify if the recognized activity is really performed.

5) ALC

A human activity has a finite life cycle, which spans from its initiation to its termination. An activity is usually initiated in response to a requirement or a stimulus. The termination of an activity is usually close to or coincides with fulfilling a certain goal.

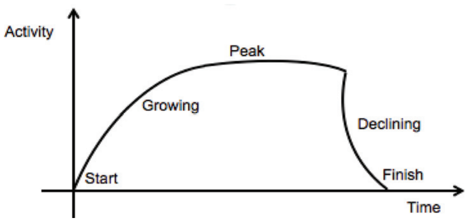

Fig. 3. Activity life cycle stages.

Tracking ALC can improve activity recognition accuracy. In Fig. 3, ALC is composed of four stages: 'starting,' 'growing,' 'declining,' and 'finishing.' When an activity starts, there may not be sufficient sensor data for making decisive conclusions. For example, if it is observed that a person holds a cup, it is not clear why the person holds the cup even though it is obvious the person may be performing an activity, such as 'drinking,' 'washing cups,' etc. In this ambiguous case, recognition decision is postponed until more sensor data is collected.

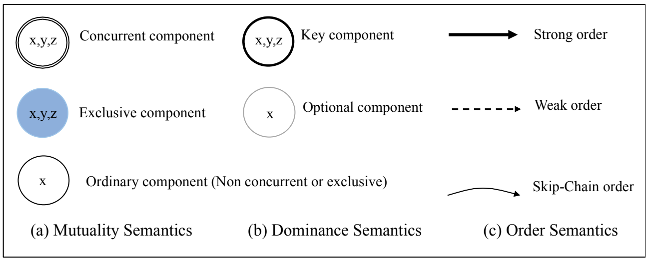

Fig. 4. Notations of activity semantics.

Fig. 4 shows the modeling notations of each semantic. Dominance semantics and mutuality semantics are represented as nodes whereas order semantics are represented as edges. Fig. 5 shows an example of an activity model designed based on the GAF and its activity semantics.

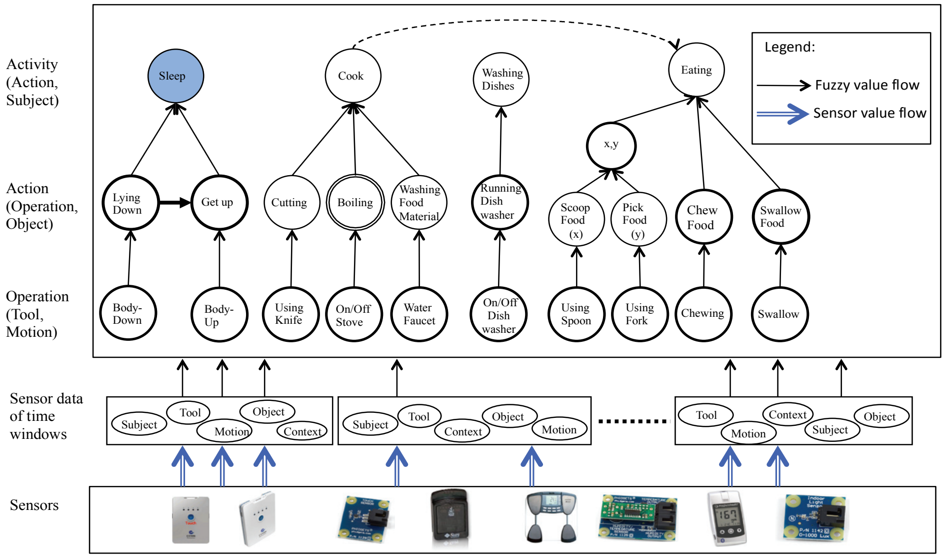

Fig. 5. An example of activity model.

4. Activity Semantic Fuzzy Logic Theory for Human Activity Recognition

An activity semantic fuzzy logic is defined by combining activity semantics with fuzzy logic [12]. There are three extensions that result from this combination: extension of fuzzy set, extension of fuzzy operators, and extension of fuzzy rules, as shown in Fig. 6.

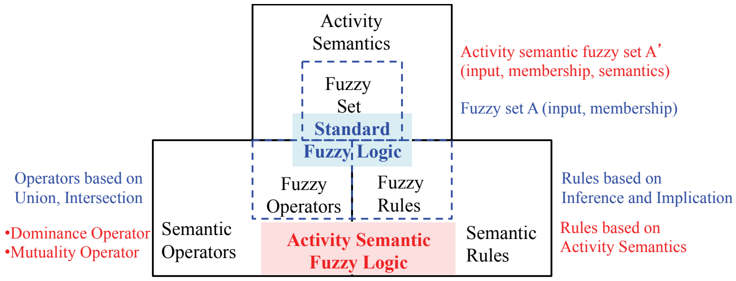

Fig. 6. Extension of fuzzy logic to semantic fuzzy logic.

4.1 Extension of Fuzzy Set to Activity Semantic Fuzzy Set

In fuzzy set F, x is an input element of a real set X. An element of semantics set S is denoted by s. A fuzzy membership value of x on fuzzy membership function u_A is denoted as $u_A(x)$ or u(x). A fuzzy set F is extended to an activity semantic fuzzy set F′ that contains activity semantics (Table 1).

Table 1. Fuzzy set and semantic fuzzy set

Set	Set expression
Fuzzy set	$F = \{(x, u_A(x)) \mid x \in X\}$
Semantic fuzzy set	$F' = \{(x,\ u_A(x),\ s) \mid x \in X, s \in S\}$

Fuzzy set F (input value, fuzzy membership value) is extended to Semantic fuzzy set F' (input value, fuzzy membership value, activity semantic) that includes an additional 'semantic' variable. For example, F = {(300, 0.3), (500, 0.5), … , (10, 0.01)} is extended to F' = {(300, 0.3, Optional), (500, 0.5, Concurrent), … , (10, 0.01, Key)}.

4.2 Extension of Fuzzy Operator to Activity Semantic Fuzzy Set Operator

Fuzzy set operator is composed of a set of T-norm and S-norm fuzzy operators, which are based on union and intersection operator. We added more fuzzy set operators that can compute fuzzy values of activity semantics.

1) Fuzzy dominance operator (operator @)

This operator computes activity dominance semantics. Fuzzy dominance operator is defined as "the values of every key component should be greater than 0 for any recognized activity". According to the defined operation, fuzzy dominance operator is defined in Table 2. In the table, x and y are operands. Their fuzzy values are u(x) and u(y). Their dominance semantic is either 'Key' or 'Optional.' We assign a fuzzy value to dominance semantics. In other words, we assign 1 to 'Key' and 0 to 'Optional' according to their dominance degree.

Table 2. Fuzzy operation for dominance semantics

x	y	Operation result (x@y)
Key	Key	(max(Key, Key), min(u(x), u(y)) =(Key, min(u(x), u(y))
Key	Optional	(Key, u(x))
Optional	Optional	(Optional, 1)

If either x or y is a key component of an activity, their dominance operation result is also a key. The fuzzy value of dominance operation is minimum fuzzy value of all key operands. If any key component has fuzzy value 0, then dominance operation will return 0.

2) Fuzzy mutuality operator (operator #)

The fuzzy mutuality operator checks concurrency or mutuality semantics between two input data. The value of concurrency is between 0 and 1, depending on concurrency. We assign 1 to 'Concurrent' and 0 to 'Exclusive' semantic. For 'Ordinary,' a fuzzy value between 0 and 1 is assigned (e.g., 0.5). Table 3 shows mutuality operation. If either x or y is exclusive, mutuality operation returns the operand that has a greater fuzzy value. Otherwise, it returns both operands.

Table 3. Fuzzy operation for mutuality semantics

x	y	Operation results (x#y)
Concurrent	Concurrent	{(1, x), (1, y)}
Concurrent (= 1)	Ordinary (= 0.5)	{(1, x), (0.5, y)}
Concurrent	Exclusive (= 0)	{(1, x)} if (u(x) > u(y)) {(0, y)} otherwise
Exclusive	Exclusive	{(0, x)} if (u(x) > u(y)) {(0, y)} otherwise
Exclusive	Ordinary	{(0, x)} if (u(x) > u(y)) {(0.5, y)} otherwise
Ordinary	Ordinary	{(0.5, x). (0.5, y)}

4.3 Extension of Fuzzy Rule to Semantic Fuzzy Rule

Fuzzy rule is based on classical implication and inference rules [13]. The difference between classical reasoning rule [14] and fuzzy rule [13] is that x and y values denote fuzzy values in fuzzy rules whereas they represent Boolean value in classical reasoning rules. In fuzzy logic, these rules are used for approximate reasoning; the resulting output is linguistic words. For example, if an implication rule (if x then y, x→y) is combined with a linguistic word 'strong' then the fuzzy rule is "if x is strong then y is strong." Then, one of inference rules is "if y is not strong, then x is not strong" because inference rules are derived from implication rule. These fuzzy implication and inference rules are extended for activity recognition; they are applied for checking activity semantics like dominance operator and mutuality operator. In activity semantic fuzzy logic, the fuzzy rule is "if an activity is clearly performed then the semantics of the activity are clear."

Table 4. Fuzzy rule for activity effect semantics

Rule	Rule expression
Implication (Rule1)	$\left(u\left(A\right)>0\right)\rightarrow\left(\text{Effect}_{(A,i)}>0\right),\exists\text{Effect}_{(A,i)}\in\text{Effect}_A$
Inference (Rule2)	$\left(\text{Rule1}\wedge\text{Effect}_{(A,i)}=0,\forall\text{Effect}_{(A,i)}\in\text{Effect}_A\right)$ $\rightarrow\left(u\left(A\right)=0\right)$

Table 4 shows an example of activity semantic fuzzy rule. Even though this example shows the fuzzy rule of effect semantics, other semantics can be also presented. The effect rule checks the semantic "if an activity is truly performed, some effects of the activity should be detected." A is an activity." $\text{Effect}_{(A,\, i)}$ is the i^{th} effect of the activity. Effect_A is a set of all effects of A.

5. APPLYING ACTIVITY SEMANTIC FUZZY LOGIC TO ACTIVITY RECOGNITION

In this section, we explain how to apply training-free fuzzy logic to activity recognition (activity recognition). Our activity recognition algorithm utilizes activity model that is of a hierarchical structure as illustrated in Fig. 5. During the initial phase of recognizing an activity, only low-level sensor data is available. From these sensor data, activity recognition algorithm should recognize performed activities by computing the fuzzy value from low layer to top layer using activity semantic fuzzy logic techniques and activity model.

5.1 Step 1: Fuzzy Membership Function

Fuzzy membership function is used at two points in the activity recognition process. First, it is used to convert sensor values to fuzzy membership values. When a sensor event is triggered, the sensor value is not initially a fuzzy value. Different sensors have a different output range. For example, a light sensor has sensor value from -10 to 10 whereas motion sensor has a value from 0 to 360. Fuzzy membership function converts these sensor values to fuzzy membership values between 0 and 1. There are popular fuzzy membership functions such as triangular or trapezoidal fuzzy membership functions [13]. Second, fuzzy membership function is used to combine the fuzzy value of an activity with linguistic terms. After computing the fuzzy value of activities, activities in top layer of activity model have a fuzzy value. This fuzzy value is a numeric value and it should be combined with a linguistic term because it is easier for people to understand. For example, if the fuzzy value of an activity is 0.5, the linguistic term will be "activity is somewhat performed" according to an example of decision index in Fig. 7.

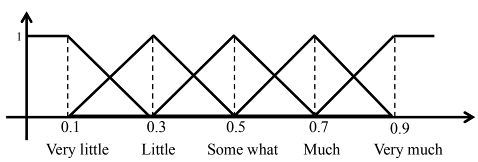

Fig. 7. Indices to decide fuzzy membership of activities.

5.2 Step 2: Computing Component Weights Based on Activity Model

Even though an activity is composed of several components in Figs. 5 and 8, their contributions to the activity vary. Some components are very crucial to determine a performed activity whereas some are not. The weight value of a component determines how strongly it impacts the activity. If the weight value of a component is high, the component has powerful influence. Many supervised machine-learning algorithms compute a weight value via training. They collect data and train the algorithm with the data. Training is a reasonable approach for certain applications if sufficient training data can be collected in a cost effective manner. However, it is difficult to apply training based algorithms to human activity recognition because human activities are very complex and it is difficult to collect sufficient data with the constraints of time and resources. In our approach, weights are computed based on the activity model. The weight (w_i) of an i^{th} child component (c_i) is determined by contribution and evidential power of the child component in Eq. (1). Because these contribution and evidential power of the child component are computed using the information in an activity model, this approach does not require training.

$$w_i = contribution_i \times evidential\_power_i \tag{1}$$

The contribution ($contribution_i$) of a child (c_i) indicates how much this child component contributes to its parent component. For instance, in Fig. 8, p1 has three children e1, e2, e3, and therefore, the contribution value of each child is 0.33.

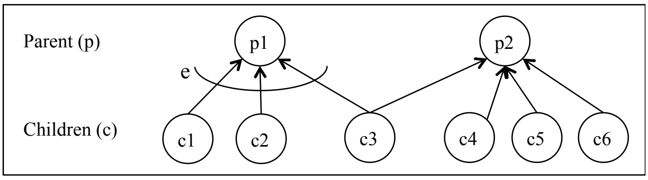

Fig. 8. Contribution and evidential_power relationship between children components and parents.

In Eq. (2), assuming that every child component contributes equally to their parent, n is the number of children components of a parent.

$$\text{contribution}_\text{i} = \frac{1}{\text{n}} \tag{2}$$

where, n is the number of incident edges to p, e(c→p, p).

Evidential power ($evidential\_power_i$) of a child (c_i) represents how much evidential information a child component has in order to determine a parent component. To illustrate, if a sensor is only used to detect a specific activity, the evidential power of the sensor data is very high. On the other hand, if a sensor is used to detect several activities, the evidential power of the sensor data will be low. In Eq. (3), m is the number of parents of a child component. For example, in Fig. 8, e3 has two parents, p1 and p2, and therefore, the $evidential\_power_i$ of e3 is 0.5.

$$\text{evidential\_power}_\text{i} = \frac{1}{\text{m}} \tag{3}$$

where, m is the of outgoing edges from c, e(c → p, c).

In the weight of a group, if multiple components are triggered at the same time, they are considered to form a group. When there is a group of children components, the weight may be computed as a group because the fact that they occur at the same time is important information to determine the performed activity. For example, in Fig. 9, if two components b and c are detected, it is highly likely that A2 was performed. When b and c are detected at different instants of time, the evidential power is 0.5 for both of them according to Eq. (3). If b and c are in a group, the evidential power of {b, c} on A2 is 1 and weight of each of them is 0.5.

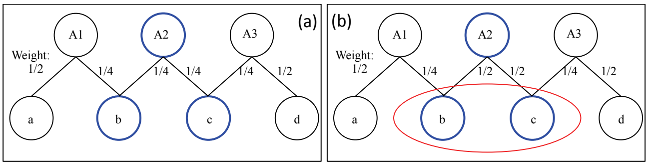

Fig. 9. Weights of group components. (a) The weight of b and c when they happened in different times. (b) The weight when b and c are detected at the same time.

5.3 Step 3: The Computation of Fuzzy Value of Activities

The fuzzy value (u(p)) of a parent component p is computed using the bounded algebraic sum of T-norm operation. The T-norm operation is the algebraic product of weight (w_i) and the fuzzy value ($u(c_i)$) of every child component c_i as shown in Eq. (4).

$$u(p) = \min\left(1,\ \sum_{i=1}^{n}\left(w_i \times u\left(c_i\right)\right)\right) \tag{4}$$

where, u(ci) is a fuzzy value of a child component and n is the number of children.

5.4 Step 4: Enforcing Semantics Using Activity Semantic Fuzzy Operators or Rules

After computing fuzzy value of activities, there may be activities whose fuzzy value is positive. Among them, some activities are truly performed. However, some activities may have a positive value even though they are not performed. Such positive values may be attributed to recognition error and uncertainty sources. To eliminate false recognition, activity semantics are evaluated. If an activity does not satisfy pre-established semantics, it is eliminated from the recognition process.

5.5 Step 5: Evaluation of ALC

Activity takes time from start to finish. Sometimes, an activity may be paused and restarted at a later time. When an activity is recognized, the life cycle of the activity is evaluated. If the activity is an on-going activity, all historical records since the activity is performed in ALC heap. When an activity is completed, then the activity is permanently recorded and it is removed from the heap. These techniques are implemented in the activity recognition system described in the next section.

6. Fuzzy Logic Based Training-Free Activity Recognition System Implementation

All aforementioned techniques, such as activity model framework, activity semantics, extended fuzzy logic for activity recognition, and activity model based weight computation are integrated in the fuzzy logic based activity recognition system in Fig. 10.

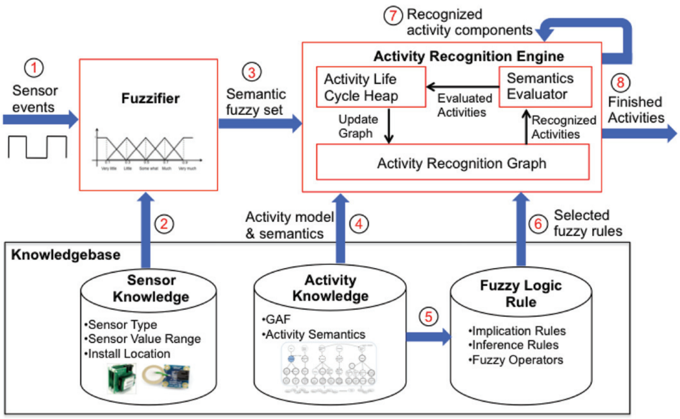

Fig. 10. The architecture of training free activity recognition system and activity recognition flow.

6.1 Activity Recognition Engine

Activity recognition engine performs fuzzy operation and processes fuzzy rules to determine performed activities using sensor data. Activity recognition engine is composed of three subsystems: activity recognition graph (activity recognition graph), semantics evaluator, and ALC heap. These subsystems interact together as shown in Fig. 11.

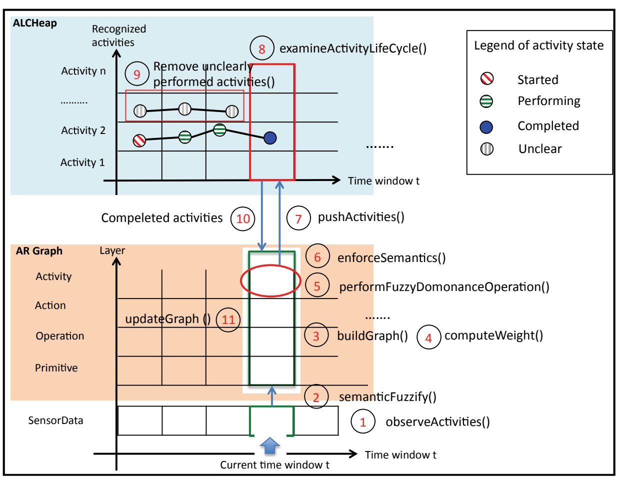

Fig. 11. Activity recognition flow of activity recognition graph, semantics evaluator, and activity life cycle heap.

Activity recognition graph is a hierarchical graph which is a subset of the activity model. When sensor event is detected, activity recognition system searches activity model and finds all related activity components and builds an activity recognition graph. After building a graph, it computes the weights for every edge and performs fuzzy operation from the bottom layers to the activity layer. After activity recognition graph computes the fuzzy values of activities, it sends all activities to ALC heap to evaluate the ALC of recognized activities as shown in Fig. 11. An activity is performed over a finite duration from activity initiation to activity completion. To determine whether an activity is completed, on-going, or paused, it is necessary to track its life cycle states.

7. Validation of Fuzzy Logic based Activity Recognition Algorithm

An experimental study has been conducted with the main objective of seeking answers to the following two questions.

- Does the fuzzy logic based, training-free activity recognition algorithm show better performance [15,16] than multi-layer neural network based activity recognition algorithm that utilizes training data?
- How much does activity semantic knowledge contributes towards increasing recognition accuracy?

To answer these two questions, we developed four activity recognition systems and compared the accuracy of four activity recognition algorithms—FL, FL+S, MLNNK, and MLNNK+S. Fuzzy logic (FL) recognizes activities using fuzzy logic only. Fuzzy logic + semantics (FL+S) recognizes both fuzzy logic and activity semantics. MLNNK recognizes activities based on multi-layer neural network. MLNNK+S recognizes activities using both multi-layer neural network and activity semantics.

7.1 Experiment Setup

Activity dataset for this validation is drawn from the domain of activities of daily living for older adults [17,18], and was collected at the Gator Tech Smart House [19-21]. Among several collected activity datasets, 'making hot tea' activity datasets that were generated by three residents for four days were used for this experiment. For collecting 'MakingHotTea,' bluetooth enabled RFID devices, Android smartphones, and several sensors were used. The bluetooth RFID reader shown in Table 5 was well suited for collecting the activity data set, its light-weight (75 g) ensures that it is wearable and it has a range of 50 cm [22], which is sufficient for the experiment. In total, 'MakingHotTea' was performed 17 times as shown in Table 6. Some of them are only partially performed to evaluate how well activity recognition algorithms deal with incomplete activities.

Table 5. Sensors and devices used to collect activity data

Sensor and device	Location
Android phone	Portable
RFID reader (bluetooth enabled) [23]	Wearable device
RFID tags	Kettle, bottles, cups, etc.
Touch sensor	Stove switch
Temperature, humidity sensors [22]	Kitchen

Table 6. Target activities and the number of cases

Activity ID	Activity	Total # of cases
1	MakingHotTea (partially performed)	11
2	MakingHotTea (completed)	6

7.2 Comparison and Analysis

According to the accuracy measurement expressions in Eqs. (5) and (6), class accuracy and time slice accuracy are computed. Class accuracy represents how accurately a target activity is recognized. Time slice accuracy represents the number of time slices that are correctly recognized out of all time slices in which the activity is performed. To compute accuracy, we measure the following four cases.

- True positive (TP): the number of activities that are performed and recognized.
- True negative (TN): the number of activities that are neither performed and nor recognized.
- False positive (FP): the number of activities that are recognized by the system, even though they are not really being performed.
- False negative (FN): the number of activities that are performed, but not recognized.

$$\text{Accuracy}_{\text{Activity}} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \tag{5}$$

$$\text{Accuracy}_{\text{Timeslice}} = \frac{\sum_{i=1}^{i=n}\text{TP}}{\text{N}} \tag{6}$$

where, N is the total number of time slices.

Table 7 and Fig. 12 compare the accuracy of activity classes for activity recognition algorithms. FL shows better accuracy for both completed and partially completed activities. When activity semantics are used, both FL+S and MLNNK+S show high accuracy. For all algorithms, partially performed activities show lower accuracy than completed activity.

Table 7. Accuracy comparison of algorithms

	Activity ID	TP	TN	FP	FN	Acc.
FL	1	2	6	0	9	0.47
	2	5	9	2	1	0.82
FL+S	1	4	6	0	7	0.59
	2	5	10	1	1	0.88
MLNNK	1	0	6	0	11	0.35
	2	5	8	3	1	0.76
MLNNK+S	1	4	6	0	7	0.59
	2	5	10	1	1	0.88

TP=true positive, TN=true negative, FP=false positive, FN=false negative, Acc.=accuracy, FL=fuzzy logic, S=semantics, MLNNK=multi-layer neural network.

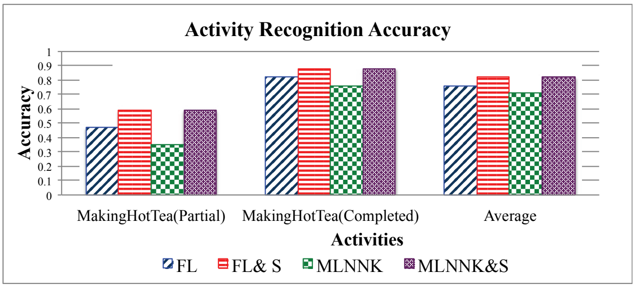

Fig. 12. The comparison of activity recognition accuracy. FL=fuzzy logic, S=semantics, MLNNK= multi-layer neural network.

Table 8 and Fig. 13 compare the time slice accuracy of activity recognition algorithms. FL and FL+S show better accuracy than MLNNK or MLNNK+S. Even though FL shows low activity recognition accuracy in Fig. 12, it shows high time slice accuracy. It means FL algorithm and FL+S algorithm recognize activities with higher confidence than MLNNK or MLNNK+S.

Table 8. Time slice accuracy

Dataset ID	Total # of time slices	FL	FL+S	MLNNK	MLNNK+S
1	13	13	13	3	13
2	9	7	7	2	7
3	2	2	2	0	2
4	6	3	6	1	4
5	3	3	3	1	3
6	7	6	7	2	6
7	4	0	0	0	0
8	3	0	0	0	0
9	4	0	4	0	0
10	3	3	3	0	0
11	1	0	0	0	0
12	3	0	0	0	0
13	6	4	6	1	4
14	18	16	16	2	16
15	2	2	2	0	2
16	5	5	5	1	5
17	2	0	0	0	0
Total	91	64	74	13	63
Acc.	1.00	0.70	0.81	0.14	0.68

FL=fuzzy logic, S=semantics, MLNNK=multi-layer neural network, Acc.=accuracy.

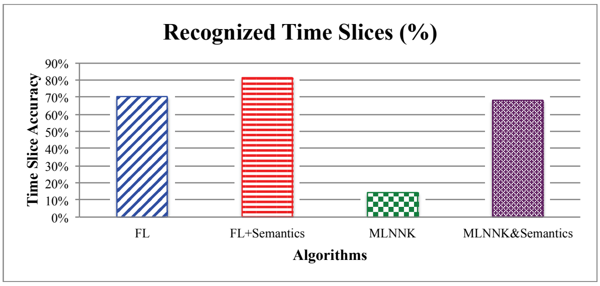

Fig. 13. The comparison of time slice accuracy. FL=fuzzy logic, MLNNK=multi-layer neural network.

8. Conclusion

We introduce a new approach for recognizing activities based on a generic model of human activities that exploits semantic knowledge of a particular domain. For instance Activities of daily living at home for elderly is one such a domain which we use in our illustrations and validation in this paper. Once a body of semantic knowledge is available, any activity in its associated domain can be defined and recognized without any required training. Even if the sensor set changes due to technology improvement or the introduction of new sensors, no training will be required. Our approach utilizes specific fuzzy logic operators and a fuzzy logic algorithm that we found to be very suitable to 1) utilizing the semantic knowledge of the domain and 2) tolerating the inherent and often complex uncertainties in activity performance. We conducted validation to show that the 'no-training' accomplishment does not come at a reduced accuracy expense. Our work clearly shows that accuracy of our approach is at least similar to or better than approaches that require training.

REFERENCES

[1] A. A. Helal, M. Mokhtari, and B. Abdulrazak, The Engineering Handbook of Smart Technology for Aging, Disability, and Independence. Hoboken, NJ: Wiley, 2008, pp. 1-26.

[2] S. Helal and E. Kim, "A practical activity recognition approach based on the generic activity framework," *International Journal of E-Health and Medical Communications*, vol. 3, no. 3, pp. 54-71, Jul. 2012.

[3] L. Constantine, "Human activity modeling: toward a pragmatic integration of activity theory and usage-centered design," in *Human-Centered Software Engineering*, A. Seffah, J. Vanderdonckt, and M. Desmarais, editors. London: Springer, 2009, pp. 27-51.

[4] T. van Kasteren, A. Noulas, G. Englebienne, and B. Kröse, "Accurate activity recognition in a home setting," in *Proceedings of the 10th International Conference on Ubiquitous Computing*, Seoul, Korea, September 21-24, 2008.

[5] E. Kim, S. Helal, and D. Cook, "Human activity recognition and pattern discovery," *IEEE Pervasive Computing*, vol. 9, no. 1, pp. 48-53, Jan. 2010.

[6] V. V. Davydov, V. P. Zinchenko, and N. F. Talyzina, "The problem of activity in the works of A. N. Leontjev," *Soviet Psychology*, vol. 21, no. 4, pp. 31-42, 1983.

[7] K. Kuutti, "Activity theory as a potential framework for human-computer interaction research," in *Context and Consciousness: Activity Theory and Human-Computer Interaction*, B. A. Nardi, Ed., Cambridge, MA: MIT Press, 1996, pp. 17-44.

[8] B. A. Nardi, "Studying context: a comparison of activity theory, situated action models, and distributed cognition," in *Context and Consciousness: Activity Theory and Human-Computer Interaction*, B. A. Nardi, Ed., Cambridge, MA: MIT Press, 1996, pp. 69-102.

[9] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257-286, Feb. 1989.

[10] C. Sutton and A. McCallum, "An introduction to conditional random fields for relational learning," in *Introduction to Statistical Relational Learning*, L. Getoor and B. Taskar, Eds., Cambridge, MA: MIT Press, 2007, pp. 93-128.

[11] H. M. Wallach, "Conditional random fields: an introduction," University of Pennsylvania CIS, Philadelphia, PA, Technical Report MS-CIS-04-21, 2004.

[12] V. Novák, I. Perfilieva, and J. Močkoř, *Mathematical Principles of Fuzzy Logic*. Boston, MA: Kluwer Academic, 1999.

[13] G. Chen and T. T. Pham, *Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems*. Boca Raton, FL: CRC Press, 2001.

[14] A. Abraham, "Rule-based expert systems," in *Handbook of Measuring System Design*, P. H. Sydenham and R. Thorn, Eds., Chichester, England: Wiley, 2005, pp. 909-919.

[15] S. A. Alvarez, "An exact analytical relation among recall, precision, and classification accuracy in information retrieval," Computer Science Department, Boston College, Chestnut Hill, MA, Technical Report BCCS-02-01, 2002.

[16] P. Lingras and C. J. Butz, "Precision and recall in rough support vector machines," in *Proceedings of the IEEE International Conference on Granular Computing*, Fremont, CA, November 2-4, 2007, pp. 654-654.

[17] W. Pentney and M. Philipose, "Sensor-based understanding of daily life via large-scale use of common sense," In *Proceedings of the 21st National Conference on Artificial Intelligence*, Boston, MA, 2006.

[18] D. Surie, T. Pederson, F. Lagriffoul, L. E. Janlert, and D. Sjölie, "Activity recognition using an egocentric perspective of everyday objects," in *Ubiquitous Intelligence and Computing, Lecture Notes in Computer Science Volume 4611*, J. Indulska, J. Ma, L. Yang, T. Ungerer, and J. Cao, Eds., Heidelberg: Springer Berlin, 2007, pp. 246-257.

[19] A. Helal, D. J. Cook, and M. Schmalz, "Smart home-based health platform for behavioral monitoring and alteration of diabetes patients," *Journal of Diabetes Science and Technology*, vol. 3, no. 1, pp. 141-148, Jan. 2009.

[20] S. Helal, C. Chao, K. Eunju, R. Bose, and L. Choonhwa, "Toward an ecosystem for developing and programming assistive environments," *Proceedings of the IEEE*, vol. 100, no. 8, pp. 2489-2504, Aug. 2012.

[21] The Gator Tech Smart House: an experimental facility at the University of Florida [Online]. Available: http://www.icta.ufl.edu/gt.htm.

[22] Ceyon Technology [Online]. Available: http://www.ceyon.co.kr/.

[23] Phidgets [Online]. Available: http://www.phidgets.com/.

Eunju Kim

Dr. Kim received a Ph.D. degree in Computer and Information Science and Engineering (CISE) from the University of Florida. Her research goals are focused on developing practical activity recognition technology that offers high accuracy and programmability required by real world applications.

Sumi Helal

Professor Helal directs the Mobile and Pervasive Computing Laboratory at the University of Florida. He is Director of the Gator Tech Smart House – an experimental research facility for researching telehealth and health telematics solutions and for supporting aging in place. Dr. Helal is Currently Associate Editor in Chief of IEEE Pervasive Computing and IEEE Computer. He can be reached at helal@cise.ufl.edu