



Context reasoning using extended evidence theory in pervasive computing environments

Daqiang Zhang^{a,*}, Minyi Guo^a, Jingyu Zhou^a, Dazhou Kang^b, Jiannong Cao^c

^a Department of Computer Science, Shanghai Jiao Tong University, Shanghai, 200240, PR China

^b Department of Computer Science, Nanjing University, Nanjing, 210089, PR China

^c Department of Computing, The Hong Kong Polytechnic University, Hong Kong

ARTICLE INFO

Article history:

Received 15 February 2009

Received in revised form

27 July 2009

Accepted 3 August 2009

Available online 7 August 2009

Keywords:

Evidence theory

Context reasoning

Pervasive computing

ABSTRACT

Most existing context reasoning approaches implicitly assume that contexts are precise and complete. This assumption cannot be held in pervasive computing environments, where contexts are often imprecise and incomplete due to unreliable connectivity, user mobility and resource constraints. To this end, we propose an approach called CRET: Context Reasoning using extended Evidence Theory. CRET applies the evidence theory to context reasoning in pervasive computing environments. Because evidence theory is limited by two fundamental problems – computation-intensiveness and Zadeh paradox, CRET presents evidence selection and conflict resolution strategies. Empirical study shows that CRET is desirable for pervasive applications.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

Pervasive computing aims to create smart environments that embed computation and communication in a manner that contextually interacts with users to ease their daily life [1,2]. Context reasoning enables pervasive applications to automatically adapt to the changeable contexts and to unobtrusively integrate users into their environments. The proliferation of pervasive applications has fostered an increasing attention to context reasoning, where contexts often refer to pieces of information that captures the characteristics of pervasive computing.

Contexts collected in pervasive computing environments are often imprecise and incomplete [3–5]. A number of sensing and computational devices, such as hand-held devices, sensor networks and Radio Frequency IDentification (RFID), can detect contexts, but they cannot provide precise contexts. For instance, in a smart meeting room, a sensor detects a context – *the location of the user is meeting room*. Usually, the sensor does not know the exact location of the user. Furthermore, sensors often miss pieces of contextual information. When the user talks in the meeting room, he or she uses many mimics to express his or her idea appropriate to the topic. In this case, contexts collected by sensors are highly

imprecise and incomplete. This is because the sensor technology, collected data and their interpretation as contexts are prone to error. The sources of errors consist of, but are not limited to, inaccurate measurement and noise from external environments [6,7]. This case is just a common example in pervasive computing environments, and it is an essential requirement for context reasoning to handle such contexts.

A variety of context reasoning approaches have been proposed to assist pervasive applications in adaption, but they mainly focus on inferring contexts from precise and complete contexts. Bayesian network [8,9], case-based [10], logic-based [11], ontology-based [12–14] and rule-based [15] have been exploited in pervasive computing environments. The typical scenario is given as below. Sensors are embedded into physical spaces to continuously collect and report their readings. Once the contexts change, context reasoning will infer hidden contexts according to sensor readings, and then notifies pervasive applications to adapt. However, most existing approaches implicitly assume that the contexts being checked are complete and precise. This assumption cannot hold in pervasive computing environments, where contexts are often imprecise and incomplete due to unreliable connectivity, user mobility, and resource constraints.

Evidence theory is a technique for reasoning the truth from pieces of information [16]. It has achieved widespread success in auditing, decision support, financial asset evaluation, process engineering and quality control. However, evidence theory is severely limited by two fundamental problems – computation-intensiveness and Zadeh paradox. In evidence theory, Dempster's

* Corresponding author.

E-mail addresses: zhangdq@sjtu.edu.cn (D. Zhang), csjcao@comp.polyu.edu.hk (J. Cao).

rule is used to combine evidence, but it is extremely computation-intensive when combining a large amount of evidence [17]. It has been proven that Dempster's rule of combination is NP-complete [18]. This problem can be partially compensated by Bayes approximation, but it only works for singleton evidence [19]. On the other hand, evidence theory does not consider conflicting evidences, which may lead to a conclusion in favor of a less probable event, a phenomenon known as the Zadeh paradox [20]. Contexts which are highly imprecise, incomplete and vary with individuals, significantly restrict evidence theory in pervasive computing environments.

In this paper, we propose an approach – CRET: Context Reasoning using extended Evidence Theory – to deal with inconsistent contexts in pervasive computing environments. CRET is applied to context collection, representation, storage and reasoning in our context-aware architecture. In order to reduce computation overhead, CRET presents a $k - l$ strategy for evidence selection to filter the less important evidence. This strategy sorts the evidence according to the weighted sum of their masses and then selects evidence with the highest beliefs. With respect to the Zadeh paradox, CRET introduces conflicting factors. To summarize, the main contributions of this paper are two-fold. (1) It introduces an approach using evidence theory for context reasoning in pervasive computing environments. It systematically shows how to apply evidence theory to context reasoning. (2) It solves the two fundamental problems of evidence theory by evidence selection and conflict resolution strategies.

The remainder of this paper is organized as follows. Section 2 introduces the related work. Section 3 briefly gives an overview of the evidence theory. Section 4 describes the CRET approach in detail and discusses evidence selection and conflict resolution strategies. Section 5 reports our empirical study and Section 6 concludes our work with future directions.

2. Related work

Context reasoning is an important technique for context-awareness and has generated a great many approaches. In general, they can be classified into case-based, logic-based, ontology-based, probabilistic and rule-based approaches. In this section, we briefly give an overview of them, as well as the advances in evidence theory.

Case-based context reasoning approaches infer contexts based on the past cases [21,10,22]. Case-based approaches, however, suffer from two problems. One is how to automatically abstract facts and generate cases, and the other is how to measure the similarity among cases. Rule-based approaches share the similar idea with case-based approaches, together with the similar limitations [15,23].

Logic-based context reasoning, involving first-order and temporal logic, is another popular reasoning scheme [8]. Gaia project used it to represent context in the form of predicate, and reasoned about high-level context based on the pre-defined rules [11]. Logic-based approaches are designed for exact reasoning so that they are not suitable for inference from imprecise and incomplete contexts. Moreover, they do not take semantics into consideration. Note that approaches based on the fuzzy logic (e.g., Type I and Type II fuzzy logic) are dedicated to fuzzy reasoning, but their systems are not easy to implement and their computation overheads are quite heavy.

Ontology-based reasoning approaches incorporate the semantics into context representation and reasoning [24]. With the support of powerful ontology software such as Stanford Protégé, IBM IODT and Maryland Swo, ontology is widely used for context representation in pervasive applications. Yet, it has limited capability in dynamically inferring contexts. It requires defining all the rules or

training a model beforehand, and a premise that all ontologies related to the specific domain must be defined already. Normal users in pervasive applications cannot satisfy these requirements due to lack of comprehensive knowledge about their domains, and thus they cannot but resort to domain experts. This incurs higher human cost and restricts the ontology application.

Probabilistic reasoning approaches make use of probability theory to predict uncertain contexts. Bayesian network can be regarded as a standard, even canonical, probabilistic reasoning technique [8,25–28]. It represents contexts by graph and probability. It requires that all hypotheses should be exclusive and exhaustive, which cannot be held in pervasive applications. Meanwhile, the Bayesian network is also criticized for its exponential computation overhead [29,30].

In contrast, evidence theory seems to be a better choice for inference from imprecise and incomplete contexts for its capability of constructing the ground truth from pieces of information. It relaxes the requirement held by the Bayesian network, and allows for probability assignment to sets or intervals. It has attracted a great deal of attention and generated many applications in various domains [31–34]. However, evidence theory suffers from two rudimentary problems – computation-intensive and Zadeh paradox. These two problems become conspicuous in pervasive applications, because contexts are not only continuously varied and generated, but also highly conflicting. In [19], Bayesian approximation was proposed to reduce the computation overhead of evidence theory, but it works for those applications that only care about singleton evidence. In [35], constant approximation cuts down the computation overhead at the cost of accuracy. Regarding the Zadeh paradox, several methods have been proposed in [36,37], but they cannot solve the Zadeh paradox completely.

3. Background

Evidence theory is a mathematical theory of evidence that constructs a coherent picture of reality through computing the probability of an event given evidence [16,38]. It is founded on the grounds of the following concepts and principles.

3.0.1. Frame of discernment

Let Θ be the Frame Of Discernment (FOD), denoting a set of mutually exclusive and exhaustive hypotheses about problem domains. Correspondingly, 2^Θ is the power set of Θ .

3.0.2. Mass

Mass stands for a belief mapping from 2^Θ to the interval between 0 and 1, represented as m . The masses of the empty set and the sum of all the subsets in power set are 0 and 1, respectively. Mass can be assigned to sets or intervals. Let \emptyset and C be the empty set and a subset of Θ . Mass m is defined as Eq. (1).

$$\begin{aligned} m : 2^\Theta &\rightarrow [0, 1] \\ m(\emptyset) &= 0 \\ \sum_{C \subseteq \Theta} m(C) &= 1. \end{aligned} \quad (1)$$

3.0.3. Belief and plausibility

The belief of a hypothesis is the sum of the beliefs for those hypotheses that are its subsets. Conversely, the plausibility of a hypothesis is the sum of all the beliefs of sets that intersect with it. Eq. (2) gives their definitions:

$$\begin{aligned} Bel(C) &= \sum_{B|B \subseteq C} m(B) \\ Pls(C) &= \sum_{B|B \cap C \neq \emptyset} m(B), \end{aligned} \quad (2)$$

where B is a subset of Θ . Bel is the degree of belief to which the evidence supports C , constituted by the sum of the masses of all sets enclosed by it. Pls denotes the degree of belief to which the evidence fails to refute C , that is, the degree of belief to which it remains plausible, i.e., the possibility that the hypothesis could possibly happen.

3.0.4. Dempster's rule

In order to aggregate the evidence from multiple sources, Dempster's rule plays a significantly meaningful and interesting part. Let C_i be a subset of Θ and $m_j(C_i)$ be a mass assignment for hypothesis C_i collected from the j -th source. For subsets $\{C_1, C_2, \dots, C_n\}$ and mass assignments $\{m_1, m_2, \dots, m_n\}$, Dempster's rule is given as:

$$\begin{aligned} \underline{m}\{C\} &= (m_1 \oplus m_2 \oplus \dots \oplus m_n)(C) \\ &= \frac{1}{K} \sum_{\tau=C} m_1(C_1) \cdot m_2(C_2) \cdot \dots \cdot m_n(C_n), \end{aligned} \quad (3)$$

where τ equals to $C_1 \cap C_2 \cap \dots \cap C_n$, and K is the normalizing constant that is defined as Eq. (4).

$$K = 1 - \sum_{\tau=\emptyset} m_1(C_1) \cdot m_2(C_2) \cdot \dots \cdot m_n(C_n). \quad (4)$$

Eqs. (3) and (4) show that Dempster's rule combines evidence over the set of all evidence. Its complexity grows exponentially with the increase of the amount of evidence. In [18], the complexity of Dempster's rule is proved as NP-complete. Meanwhile, once some probabilities of evidence are zero, Dempster's rule will get a wrong inference that a less probable event is regarded as the most probable event that could happen. This phenomenon is named for Zadeh paradox.

4. Context reasoning using extended evidence theory in pervasive computing environments

We present a context-aware architecture for pervasive applications, where CRET: Context Reasoning using extended Evidence Theory, is applied. We first introduce our context-aware architecture and then describe CRET in detail, followed by discussions.

4.1. System model

Fig. 1 illustrates our context-aware architecture, which is a hierarchical module consisting of context providers, a manager and consumers. The context providers gather context data on the environments from sensors. Note that contexts may come from applications, and our context collection is capable of collecting contexts at intervals or on demand. The context manager preprocesses contexts and responds to requests from adapters and services. After the context information is generated, the context manager will format it and remove inconsistencies in the preprocessing step, and then move into the reasoning step where hidden

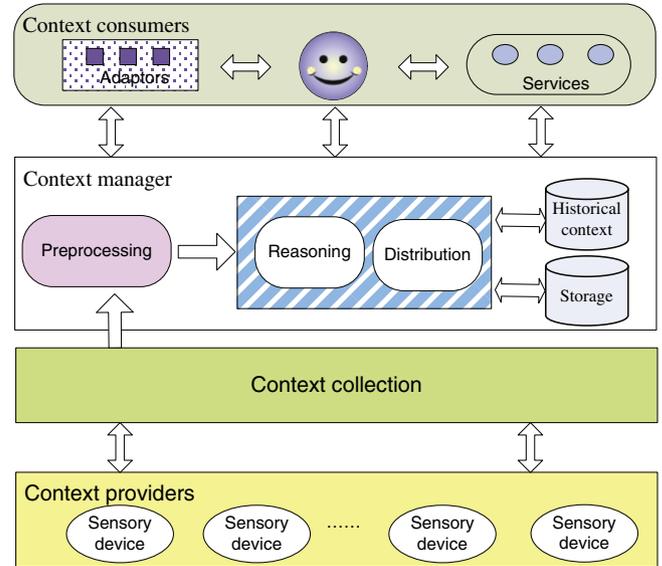


Fig. 1. Context-aware architecture. Context-awareness is provided by context collection, preprocessing, reasoning and distribution.

contexts are derived from present and historical contexts. A historical context database is introduced to store the past contexts, because contexts that are updated frequently increase in size considerably. Finally, the context manager notifies adapters and services, and distributes contexts to them according to their specific privacy and security requirements and their resource constraints. We are implementing a prototype to demonstrate our context-aware architecture. We employ TinyOS 2.0, Micaz and Microsoft SQL server to collect, preprocess and store contexts. Our prototype currently supports devices with Bluetooth, Wi-Fi and WLAN function.

It is noteworthy to point out that CRET is applied to context collection, reasoning and storage in the context manager. This enables the context manager to handle imprecise and incomplete contexts. The goal of this study is to provide a context-aware architecture with context reasoning functionality. The current version of CRET is a centralized approach, where a static node is selected to centrally control the system. From here on, we describe the process of applying evidence theory to pervasive applications and then extend evidence theory to overcome its two significant problems.

4.2. CRET model

We model the sensor-driven pervasive applications as a hierarchical structure, consisting of sensors, objects, and activity layers [39]. Sensors are deployed to monitor objects in the sensor layer. All objects are in the object layer, where some objects are monitored by sensors. Activities are in the activity layer, which can be inferred according to the contexts of objects. In order to formally describe our model, we define $S = \{s_1, s_2, \dots, s_n\}$ as the set of sensors, $O = \{o_1, o_2, \dots, o_n\}$ as the set of objects, $A = \{a_1, a_2, \dots, a_n\}$ as the set of activities, SO as simple objects, monitored by sensors directly, DO as deduced objects, obtained from SO , CO as composite objects, comprised of SO or DO , E as the set of evidence, and D as discount rate, reflecting the reliability of sensor readings. The system infers the activity candidate set and then makes a decision using Dempster's rule.

Fig. 2 illustrates our system model, where o_5 and o_6 are deduced object and composite object, and lines with various arrows denote relationships among sensors, objects and activities. Objects o_1, o_2, o_3 and o_4 are monitored by sensors s_1, s_2, s_3 and s_4 , respectively. Object o_5 is deduced from o_1 , and o_6 is made up of objects o_2 and

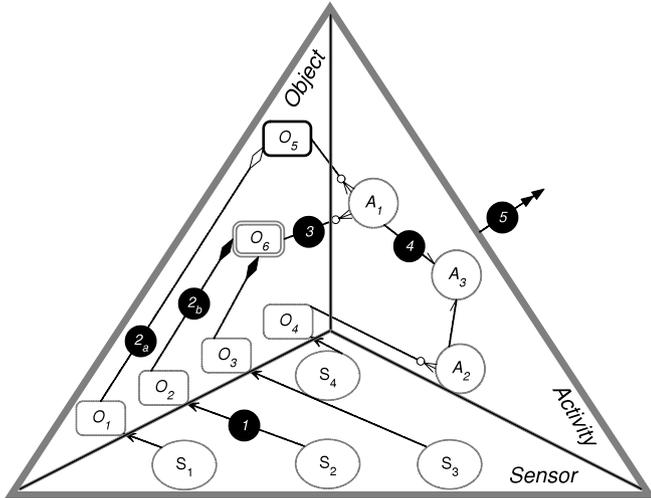


Fig. 2. System model for CRET.

o_3 . Suppose a scenario that a sensor is deployed to monitor the door of a refrigerator where coffee, milk and tea are stored. When the door is open, users make a drink by selecting coffee, milk or tea, or their combinations as simple a_1 or a_2 , or complex activity a_3 . In this case, the refrigerator is type of object o_4 , coffee, milk and tea are type of object o_5 and their combinations are type of object o_6 .

In Sections 4.3–4.5, we describe how to apply CRET to this scenario. The process of applying CRET consists of (1) propagating evidence in the sensors layer, (2) propagating evidence in the objects layer, and (3) recognizing activities. In Section 4.6, we solve the two problems of CRET - intensive computation and Zadeh paradox.

4.3. Propagating evidence in the sensors layer

Sensors are vulnerable and vary drastically with environments, which leads to unreliable sensor readings. Therefore, it is important to take into account the reliability of sensors in the evidence aggregation process. As proposed by Shafer [16], this reliability can be controlled by the discount rate, which is derived from the sensor reliability model [31]. In CRET, the sensor discount rate is incorporated, given as Eq. (5):

$$m^r(C) = \begin{cases} (1-r)m(C) & C \neq \emptyset \\ r + (1-r)m(\emptyset) & \text{otherwise,} \end{cases} \quad (5)$$

where r is the sensor discount rate with its value between 0 and 1. When r is 0, the sensor is completely reliable; when r is 1, the sensor is absolutely unreliable.

In the beginning, sensors s_1, s_2, s_3 and s_4 are installed and working with discount rates as 0.02, 0.02, 0.1 and 0.2. Evidence on sensor nodes is represented by masses as: $m_{s_1}(s_1) = 1, m_{s_2}(s_2) = 1, m_{s_3}(\neg s_3) = 1$ and $m_{s_4}(s_4) = 1$. Then, the discounted masses are calculated as:

$$\begin{aligned} m^r(s_1) &= 0.98, & m^r(s_1, \neg s_1) &= 0.02 \\ m^r(s_2) &= 0.98, & m^r(s_2, \neg s_2) &= 0.02 \\ m^r(\neg s_3) &= 0.9, & m^r(s_3, \neg s_3) &= 0.1 \\ m^r(s_4) &= 0.8, & m^r(s_4, \neg s_4) &= 0.2. \end{aligned}$$

There is a relationship between sensors and their associated objects. For example, given the frames \emptyset_a and \emptyset_b of sensor s_a and its associated object o_b , the relationship between sensor s_a and object o_b denotes the evidence propagation from sensors to objects layers. CRET associates objects with sensors by maintaining a compatible relationship between them, which is defined as evidential mapping by Eq. (6):

$$\Gamma : m(o_b) = f(s_a \rightarrow o_b), \quad (6)$$

Table 1
Evidential mapping.

Relationship	Evidence value
$\{o_1\} \rightarrow \{o_5\}$	0.9
$\{\neg o_1\} \rightarrow \{\neg o_5\}$	1.0
$\{o_1\} \rightarrow \{o_5, \neg o_5\}$	0.1
$\{o_1, \neg o_1\} \rightarrow \{o_5, \neg o_5\}$	1.0
$\{o_4\} \rightarrow \{a_2\}$	0.3
$\{\neg o_4\} \rightarrow \{\neg a_2\}$	1.0
$\{o_4\} \rightarrow \{a_2, \neg a_2\}$	0.7
$\{o_4, \neg o_4\} \rightarrow \{a_2, \neg a_2\}$	1.0

where f is a mapping function, propagating the evidences from sensor s_a to object o_b . For objects o_1, o_2, o_3 and o_4 in Fig. 2, their masses are computed as:

$$\begin{aligned} \Gamma : m(o_1) &= m^r(s_1) = 0.98 \\ \Gamma : m(o_1, \neg o_1) &= m^r(s_1, \neg s_1) = 0.02 \\ \Gamma : m(o_2) &= m^r(s_2) = 0.98 \\ \Gamma : m(o_2, \neg o_2) &= m^r(s_2, \neg s_2) = 0.02 \\ \Gamma : m(\neg o_3) &= m^r(s_3) = 0.9 \\ \Gamma : m(o_3, \neg o_3) &= m^r(s_3, \neg s_3) = 0.1 \\ \Gamma : m(o_4) &= m^r(s_4) = 0.8 \\ \Gamma : m(o_4, \neg o_4) &= m^r(s_4, \neg s_4) = 0.2. \end{aligned}$$

In this step, the evidence is propagated to all the sensors and objects. With the help of the mapping function, the masses of sensors are transferred to the objects layer.

4.4. Propagating evidence in the objects layer

This step intends to propagate evidence in the objects layer. We collect evidence from the observations illustrated in Table 1. Note that evidence $\{o_1\} \rightarrow \{o_5\}$ refers to that a 0.9 confidence about object o_5 when we observe object o_1 . We take simple object o_4 , deduced object o_5 and composite object o_6 to explain this step.

4.4.1. Calculating masses for the deduced objects: $DO \leftarrow E$

Although the deduced objects are not monitored directly by sensors, evidential mapping is capable of calculating their masses from their evidence. For instance, evidential mapping propagates masses from objects o_1 to o_5 as:

$$\begin{aligned} m(\{o_5\}) &= m(\{o_1\}) * m(\{o_1\} \rightarrow \{o_5\}) \\ &= 0.882 \\ m(\{o_5, \neg o_5\}) &= m(\{o_1\}) * m(\{o_1\} \rightarrow \{o_5, \neg o_5\}) \\ &\quad + m(\{o_1, \neg o_1\}) * m(\{o_1, \neg o_1\} \\ &\quad \rightarrow \{o_5, \neg o_5\}) \\ &= 0.118. \end{aligned}$$

4.4.2. Calculating masses for the composite objects: $CO \leftarrow E$

This step aims to propagate evidence to the composite objects. This is also achieved by evidential mapping and is much more complex than those of the deduced objects. It includes two substeps: a) calculating masses to the sets of objects that comprise the composite objects, and b) calculating masses of the composite objects. In the first step, CRET computes the masses of these sets using Eq. (6).

$$\begin{aligned} \Gamma : m_1(\{o_2, o_3\}) &= m(\{o_2\}) = 0.98 \\ \Gamma : m_2(\{o_2, o_3\}) &= m(\{o_3\}) = 0.9 \\ \Gamma : m_1(\{o_2, o_3, \neg(o_2, o_3)\}) &= m(\{o_2, \neg o_2\}) = 0.02 \\ \Gamma : m_2(\{o_2, o_3, \neg(o_2, o_3)\}) &= m(\{o_3, \neg o_3\}) = 0.1. \end{aligned}$$

Thus, we get different beliefs for observation combinations from two sources, e.g., beliefs of $m_1(\{o_2, o_3\})$ and $m_2(\{o_2, o_3\})$ from readings of sensors s_2 and s_3 . How to generate the masses of the

composite objects is an interesting job. We present an aggregation mechanism by using the weighted sum, defined as Eq. (7):

$$m(C) = m_1 \hat{\oplus} m_2 \hat{\oplus} \dots \hat{\oplus} m_n(C)$$

$$= \frac{\sum_{i=1}^n w_i \cdot m_i}{\sum_{i=1}^n m_i}, \quad (7)$$

where C is a subset of Θ , w_i is the weight for $m_i(A)$ and $\sum_{i=1}^n w_i$ equals to 1. Let r_i be the discount rate for mass m_i . The weight of w_i is calculated by Eq. (8).

$$w_i = \frac{r_i}{\sum_{i=1}^n r_i}. \quad (8)$$

Theorem 1. *The mass $m(C)$ that is calculated by Eq. (7) is a proper mass.*

Proof. A proper mass means that a mass must satisfy all the requirements given by Eq. (1). Because m_i is less than 1, the value of $\frac{\sum_{i=1}^n w_i \cdot m_i}{\sum_{i=1}^n w_i}$ is between 0 and 1. When C is an empty set, the value of $m(C)$ is equal to 0. Suppose we have l subsets of Θ . The sum of $m(C_j)$ is calculated as $\frac{\sum_{j=1}^l \sum_{i=1}^n m_{ji}}{\sum_{i=1}^n m_i}$ that is equal to $\frac{\sum_{i=1}^n \sum_{j=1}^l m_{ij}}{\sum_{i=1}^n m_i}$. The value of $\sum_{j=1}^l m_{ij}$ is m_i so that the sum of all the subsets of Θ equals to 1. Therefore, $m(C)$ is a proper mass. \square

In our case, parameters w_2 and w_3 are computed as 0.6 and 0.4, and ϖ equals to the value of $w_2 * m_1(\{o_2, o_3\}) + w_3 * m_2(\{o_2, o_3\})$. Eq. (9) shows the evidential mapping for $m(\{o_6\})$. In the same way, we get the mass value of $m(\{o_6, \rightarrow o_6\})$.

$$m(\{o_6\}) = m(\{o_2, o_3\})$$

$$= \frac{\varpi}{w_2 + w_3} = 0.948. \quad (9)$$

4.4.3. Generating an activity candidate set from objects: $A \leftarrow O$

So far, we have generated an activity candidate set in which users may do one or more activities at a time. *Although knowing precisely what the users want is impossible, observations show that most routine tasks are predictable.* In fact, an object-activity mapping table is collected in CRET that conforms closely to the reality of how users work. This mapping table varies with each scenario and can be collected from the observations. For Fig. 2, activity a_1 obtains the masses propagated by objects o_5 and o_6 according to the object-activity mapping table. Then its mass is calculated as:

$$m_1(\{a_1\}) = m(\{o_5\}) = 0.882$$

$$m_2(\{a_1\}) = m(\{o_6\}) = 0.948$$

$$m_1(\{a_1, \rightarrow a_1\}) = m(\{o_5, \rightarrow o_5\}) = 0.118$$

$$m_2(\{a_1, \rightarrow a_1\}) = m(\{o_6, \rightarrow o_6\}) = 0.052.$$

For the same reason, the mass of activity a_2 is calculated as Eq. (10). At this point, CRET has generated an activity candidate set and is ready for user activity recognition in the next step.

$$m_{a_2}(\{a_2\}) = m_{o_4}(\{o_4\}) * m(\{o_4\} \rightarrow \{a_2\})$$

$$= 0.24$$

$$m_{a_2}(\{a_2, \rightarrow a_2\}) = m_{o_4}(\{o_4\}) * m(\{o_4\} \rightarrow \{a_2, \rightarrow a_2\}) + m_{o_4}(\{o_4, \rightarrow o_4\}) * m(\{o_4, \rightarrow o_4\} \rightarrow \{a_2, \rightarrow a_2\})$$

$$= 0.76. \quad (10)$$

4.5. Recognizing user activities: $a_i \leftarrow A$

This step focuses on recognizing user activities from the activity candidate set. For example, activity a_1 is chosen because of its bigger mass than that of other activities. Observations in previous

Table 2
An example of activity candidate set.

Activity	m_1	m_2	\underline{m} - Final result
$\{a_1\}$	0.40	0.20	?
$\{a_2\}$	0.30	0.20	?
$\{a_3\}$	0.20	0.30	?
$\{a_1, a_2\}$	0.05	0.15	?
$\{a_2, a_3\}$	0.04	0.10	?
$\Theta = \{a_1, a_2, a_3\}$	0.01	0.05	?

steps reveal that the original accuracy of sensors has a significant impact on the final results. Two methods are available to avoid the influence of sensor reliability. One method is to use powerful sensors to improve accuracy, but this increases cost and prevents pervasive computing. The other method is to use multiple sensors to monitor the same objects or sample the same context several times. In most cases, the latter method is preferred for pervasive applications because it inflicts less cost.

To describe clearly the mechanism of activity recognition, we extend the example used in Section 4.4.3. Suppose that we collect the evidence for the same scenario and process it by the steps mentioned above (see Table 2). According to the Eq. (4), the normalizing constant K is calculated as

$$K = 1 - \sum_{B \cap C = \emptyset} m_1\{B\} \cdot m_2\{C\}$$

$$= 1 - \{m_1\{a_1\} \cdot m_2\{a_2\} + \dots + m_1\{a_2, a_3\} \cdot m_2\{a_1\}\}$$

$$= 1 - \{0.4 * 0.2 + \dots + 0.04 * 0.2\}$$

$$= 0.477.$$

Note that the calculation of the normalizing constant is computed over the entire set of evidence. When the amount of evidence and sources increase, Dempster's rule requires exponential time to calculate the normalizing constant. Existing work [18] has proved that the complexity of Dempster's rule is NP-complete. Moreover, when some evidence is zero, Dempster's rule will get wrong results, which is well-known as the Zadeh paradox. According to the Eq. (3), the combination mass of activity a_1 is computed as

$$\underline{m}\{a_1\} = m_1 \oplus m_2(\{a_1\})$$

$$= \frac{\sum_{B \cap C = a_1} m_1(B) \cdot m_2(C)}{K} = 0.3606.$$

In the same way, we calculate the combined masses $\underline{m}\{a_2\}$, $\underline{m}\{a_3\}$, $\underline{m}\{a_1, a_2\}$, $\underline{m}\{a_2, a_3\}$ and $\underline{m}\{\Theta\}$ as 0.3795, 0.2201, 0.0241, 0.0147 and 0.0010, respectively. After all the combination masses have been obtained, applications will compute the beliefs using Eq. (2) and then recognize the activities. Correspondingly, activity a_2 in Table 2 is selected as the prediction. Up to now, all the steps of applying CRET to pervasive computing environments have been described. However, two problems are still not solved – the intensive computation that arises from calculating the normalizing constant and combined masses [16], and the Zadeh paradox that emerges from conflicting evidences [20].

4.6. Evidence selection and conflict resolution strategies

In order to solve the problems of evidence theory, we present two strategies (i.e., evidence selection and conflict resolution to the Zadeh paradox).

4.6.1. Evidence selection strategy

The calculation for the normalizing constant K in Dempster's rule is time-consuming because it requires traversing all the

evidence and masses. We find that an observation illustrated as Lemma 1 can reduce the operations for the normalizing constant to $|\Theta|$. Note that Dempster's rule, which uses the Eq. (4) to calculate the normalizing constant, is appropriate for calculating the combined mass for any one hypothesis without calculating all the combined masses.

Lemma 1. The normalizing constant K can be computed from all the masses that are to be normalized.

Proof. Let m_1, m_2, \dots, m_n be n masses that are to be normalized. Given that the normalizing process is to normalize all the masses, the normalizing constant K must equal to $\sum_{i=1}^n m_i$. \square

The other computation overhead of CRET emerges from the calculation of the combined masses. Suppose we calculate the combined mass for the evidence C by calculating its unnormalized mass and then normalizing it. The total number of operations required is proportional to $\log(|C|) + \log(|\Theta|)$ for ordered lists, but is exponential in $|\Theta|$ for n -dimensional array representation. Because the decisions made by pervasive applications, in most cases, rely on the most related contexts, CRET employs a process of selecting evidence. This process consists of two substeps.

First, CRET defines the importance of evidences by t_i that is given as:

$$t_i = \sum_{j=1}^n w_j \cdot m_j, \quad (11)$$

where j is the number of masses and w_j is the weight of mass m_j for evidence i . Consider that the masses from multiple sources have different impact on the decision-making for pervasive applications. CRET is capable of tuning w_j to emphasize the most important masses.

Second, CRET selects the most related evidence by $k-l$ strategy. Let n be the amount of evidence, k be the minimum amount of evidence to be kept, l be the maximum amount of evidence to be kept, η be the threshold on the sum of masses of selected evidence, Σ be the sum of masses of selected evidence, ϑ be the number of selected evidence and LSP be the least small probability event. Lines from 5 to 17 in Algorithm 1 are $k-l$ selection strategy for evidence selection. When l is equal to k , the selection strategy turns into the top K selection strategy. Note that the step of randomized quicksort in the $k-l$ strategy is just marking the order of evidence according to the t_i and does not swap evidence orders during the sorting process.

Algorithm 1 optimization selection algorithm

```

1: procedure SELECTION
2:   input:  $e$ : evidence set
3:   output:  $e'$ : selected  $k-l$  evidences
4:    $a_i$ : the prediction result
5:   //  $k-l$  selection strategy
6:   // calculate the weighted sum of masses
7:   for  $i \leftarrow 1, n$  do
8:     compute  $t_i$ 
9:     //  $rst$ : result of all  $t_i$ 
10:     $rst \leftarrow t_i$ 
11:   end for
12:   randomized quicksort  $e$  based on  $t$ 
13:   // select top  $k$  evidence
14:    $e' \leftarrow e(k)$ 
15:   while  $(\Sigma > \eta \ \&\& \ \vartheta < l)$  do
16:      $e' \leftarrow$  the evidence with the next highest mass
17:   end while
18:   // conflict resolution strategy
19:   identify  $LSP \leftarrow (rst, e')$ 
20:   minimize  $Pls(LSP, \delta)$ 
21:    $a_i \leftarrow$  recognize activities
22: end procedure

```

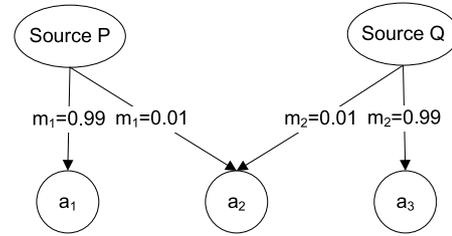


Fig. 3. A well-known example of the Zadeh paradox.

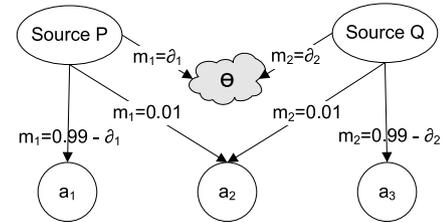


Fig. 4. Introducing conflicting factors for the Zadeh paradox.

4.6.2. Conflict resolution to Zadeh paradox

The Zadeh paradox refers to the situation that Dempster's rule becomes problematic when combining conflicting evidence whose sum masses are zero [20]. Fig. 3 illustrates a well-known example of the Zadeh paradox, where activity a_2 is a low probability event supported by little evidence, while activities a_1 and a_3 are strongly supported by some evidence. Dempster's rule, however, implies that activity a_2 with belief as 1 should be absolutely believed, and activities a_1 and a_3 get the beliefs as 0. This result is counter-intuitive. In pervasive computing environments, the evidence is often highly imprecise and incomplete and thus the Zadeh paradox is more likely to happen. In our experiments, we find that the Zadeh paradox has a great impact on the performance of CRET.

According to [17,37], all the masses are conditional beliefs (i.e., $m(x|e) = m(x)$) and thus evidence are partially believed. In essence, it is the masses of activities a_1 and a_3 that cause the unsatisfactory results. In order to remove the Zadeh paradox, we introduce conflicting factors. Fig. 4 illustrates the assignment of conflicting factors where they are added to the FOD, which affects the degree of beliefs for all hypotheses.

Definition 1. Let ε be the measurement for the Zadeh paradox, and set its value equal to the minimum combined mass of the evidence.

Definition 1 shows the measurement for the Zadeh paradox. The less the ε , the better the performance of solving the Zadeh paradox. Theorem 2 proves that conflicting factors can reduce the influence of the Zadeh paradox.

Theorem 2. Conflicting factors can decrease the value of ε for that well-known example of the Zadeh paradox.

Proof. In Fig. 3, the combined mass $\underline{m}\{a_1\}$ is $\frac{0.99-\delta_2}{1-\delta_2+(\delta_2+0.0001)/\delta_1}$ according to the Dempster's rule. Given a specific δ_2 , we infer that mass $\underline{m}\{a_1\}$ is monotonic increasing. For the same reason, we infer that the combined mass $\underline{m}\{a_3\}$ is monotonic increasing. According to the principle that the sum of all combined masses is a sum up to 1, the combined mass for activity a_2 decreases. That means measurement ε becomes smaller. Thus, conflicting factors reduce the influence of the Zadeh paradox. \square

Table 3
Result of Zadeh paradox.

Approach	Evidence Theory	Yager	Smet	CRET
$\{a_1\}$	0	0	0	0.49
$\{a_2\}$	1.0000	0.0001	0.0001	0.015
$\{a_3\}$	0	0	0	0.49
\emptyset	0	0	0.9999	0
$\Theta = a_1, a_2, a_3$	0.00	0.9999	0	0.005

The conflict resolution strategy is composed of three steps. It first identifies the *LSP* from the result of selection strategy. Then, it minimizes the plausibility function for *LSP* to calculate conflicting factors. Finally, it recognizes user activities by using Dempster's rule. Conflicting factors for conflicting evidence are defined as Eq. (12), where $m'(x|e)$ is a new mass assignment and $m'(\Theta)$ is the mass assignment for the FOD.

$$m'(x|e) = (1 - \delta) \cdot m(x|e), \quad (12)$$

$$m'(\Theta) = \delta * m(x|e) + m(\delta).$$

Given that the plausibility describes the degree of belief to which the evidence fails to refute a specific hypothesis, we infer that the less the plausibility for the least small probability event (e.g., activity a_2) is, the better result Dempster's rule can achieve. Therefore, we get an optimization problem as Eq. (13).

$$\begin{aligned} &\text{Minimize: } Pls(LSP, \delta) \\ &\text{subject to: } \delta \in (0, 1). \end{aligned} \quad (13)$$

Take Table 2, for example. The values for δ_1 and δ_2 are derived as 0.01 by minimizing the plausibility function for activity a_2 . Table 3 illustrates the final result, denoting that activities a_1 and a_3 are most likely to happen. CRET achieves the best results among all the solutions. Yager [36] and Smet [37] just identify the least probable event without pointing out the most probable event. We also test that these work over other cases of the Zadeh paradox, which shows that the results of CRET are the closest to the ground truth.

4.7. Discussion

Comments. Evidence theory has achieved widespread success in various domains. In this paper, we have shown how to apply it to pervasive applications by extending it so as to reduce the computation overhead and avoid the Zadeh paradox. Note that the $k - l$ evidence strategy is unique in complexity reduction. It neither defines any mass function nor gives new combination rule (e.g. [40]) but it remarkably reduces the size of the evidence set, and thus accelerates the evidence combination. Furthermore, it differentiates the evidence from various sources by using a set of weights that can be specified according to application requirements. With respect to the Zadeh paradox, Dempster's rule does not consider the conflicting evidence. Without conflicting evidence, Dempster's rule performs quite well, which is the reason that we extend it rather than propose a new combination rule.

Complexity. For $k - l$ evidence selection strategy, the time complexity is $O(n)$, where n is the amount of evidence. The space complexity is also $O(n)$. For the conflict resolution strategy, the time complexity is $O(\vartheta)$, where ϑ is the amount of evidence used in Dempster's rule. Because ϑ is much less than n , CRET is scalable even for a large amount of evidence.

5. Evaluation

In order to evaluate our proposed approach, we carried out a series of experiments. In particular, we tried to answer the following questions:

- What is the overall performance of CRET? Does it work better than other approaches in pervasive computing environments?
- How do two problems of evidence theory (i.e., intensive computation and Zadeh paradox) affect the performance of CRET?

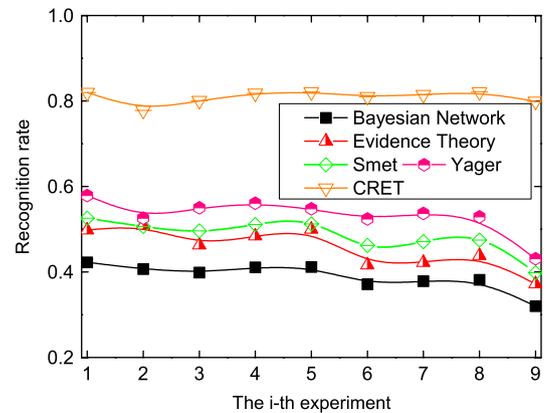


Fig. 5. Recognition rate of recognizing users' activities.

5.1. Experimental settings

We extracted ten users' activities from *Houser_n* project that is a joint project for studying and designing strategies in real living environments [41]. We selected singleton activities and complex activities together from *PLCouple1* dataset as the test activities. This dataset provides log files recording the real activities, videos and experimental data. The selected singleton activities are *watching TV*, *walking in the kitchen*, *standing still*, *walking in the live room*, *walking with carrying load*, *kneeling*, *running*, *squatting*, *bending in the dining area* and *bending in the kitchen*. The combinations of these singleton activities are possible complex activities that may be executed by users. According to the statistics, we collected the contexts information (e.g., the evidence and relationships between objects and activities). Due to unavailability of discount rates of all sensors, we assume all sensors work properly with zero discount rate. We first extracted the activity set that consists of singleton and complex evidence. Then, we did experiments nine times, and each time we used different months' data.

We select a recognition rate as the evaluation metric in our experiments. The recognition rate is defined as the ratio of the number of correctly recognized activities to the total number of activities. We run the program on a Windows XP (SP3) system with 3.0 GHz Pentium IV CPU and 1.5 G memory.

5.2. Overall performance

In this section, we conducted a series of experiments to evaluate Bayesian network and CRET. We select Bayesian network [42], because it is regarded as a typical or even canonical approach for inferring contexts in pervasive computing environments [8, 9,25–28]. Note that rule-based and logic-based approaches are designed for exact reasoning, which lead to working improperly when handling imprecise and incomplete contexts.

Fig. 5 shows the results. The x-axis is the i -th experiment. CRET outperforms the others approaches by about 28.65% in terms of recognition rate. Original evidence theory achieves a better result than the Bayesian network. This is because the evidence used in experiments is not highly conflicting. Due to two reasons, Bayesian network cannot achieve a high recognition rate. We take Table 2, for example, to explain the reasons that Bayesian network fails to handle such evidence.

There are six hypotheses $\alpha, \beta, \gamma, \eta, \xi$ and Θ , denoting $a_1, a_2, a_3, \{a_1, a_2\}, \{a_2, a_3\}$ and $\{a_1, a_2, a_3\}$, respectively. First, α relates to η and Θ , while η, ξ and Θ correlate each other. They are not exclusive and exhaustive (e.g., $\{a_1, a_3\}$ is missing). Second, causal Bayesian network cannot establish the hierarchical relationships among hypotheses when constructing its structure. In fact,

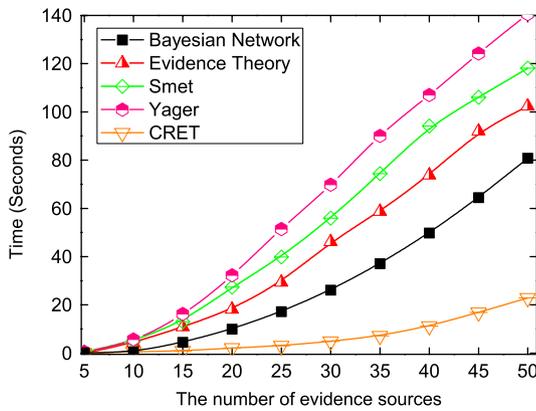


Fig. 6. Scalability with $k - l$ evidence selection strategy.

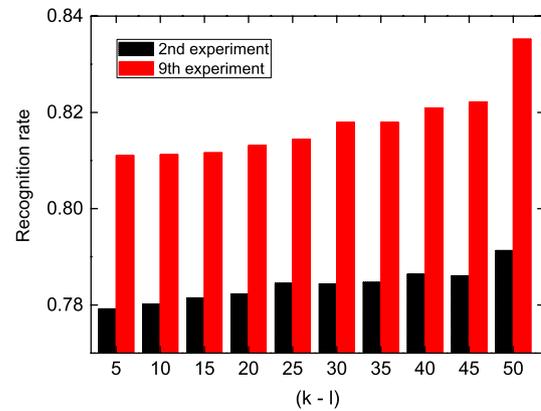


Fig. 7. Sensitivity of the $k - l$ selection strategy.

Table 4
Results for the example of Zadeh paradox.

Approach	Evidence	Yager	Smet	CRET
$\{a_1\}$	0	0	0	0.49
$\{a_2\}$	1.0000	0.0001	0.0001	0.015
$\{a_3\}$	0	0	0	0.49
\emptyset	0	0	0.9999	0
$\Theta = a_1, a_2, a_3$	0.00	0.9999	0	0.005

hypotheses α , β and γ are in the same level with η , ξ and Θ . Therefore, the two requirements in Bayesian network cannot be satisfied, which leads to its failure.

5.2.1. Scalability with selection strategy

Scalability is an important concern for context reasoning approaches. However, CRET is computation-intensive, which significantly affects its performance. Therefore, we present the selection strategy to reduce its computation overhead. To evaluate the scalability of CRET, we conducted the experiments by simulation. In our experiments, we increased the amount of evidence that is combined from 5 to 50. We simply set all the weights for masses as the same when using $k - l$ selection strategy. The other parameters are set as $k = 10$, $l = 40$ and $\delta = 0.6$.

Fig. 6 illustrates the results of scalability experiments. With the growth of the amount of evidence that is combined, the times for all approaches increase. Among all approaches, CRET is most efficient, which is guaranteed by selection strategy. According to the Fig. 5, we infer that the selection strategy really assists CRET in reducing computation overhead without loss of overall performance. Compared with the original evidence theory, Bayesian network needs less time, but its computation overhead is still heavy. Although Yager [36] and Smet [37] partially solve the Zadeh paradox, they require more time than original evidence theory. This is because they spend much time identifying conflicting evidences and solving the conflict.

5.2.2. Performance regarding Zadeh paradox

The Zadeh paradox has a significant impact on the performance of evidence theory. Yager and Smet proposed two methods to solve it. In order to compare our algorithm with them, we conducted the experiments from two aspects. We first checked whether they work for Table 2. Then, we evaluated them on the dataset mentioned in Section 4.6.2.

Table 4 illustrates the combination results for the example of Zadeh paradox. CRET achieves the best results among all solutions. The work of Yager and Smet just identifies the least probable event without pointing out the most probable events. We also tested them over other cases of Zadeh paradox (Table 2 is an

extreme case of Zadeh paradox), which shows that the results of CRET are most close to the truth. This is achieved by the conflict resolution strategy. Fig. 5 shows the results of recognition rate that Smet, Yager and CRET achieved. The solutions of Smet and Yager outperform the original evidence theory in terms of recognition rate. This is because they solve some cases that do not contain the highly conflicting evidence as Table 2.

Fig. 6 illustrates their scalability performance. With the growth of the amount of evidence to be combined, CRET requires the least time, while Yager and Smet approaches require exponential time. This is achieved by the evidence selection strategy, and previous work does not consider scalability too much.

5.3. Sensitivity of the $k - l$ selection strategy

CRET is affected by several parameters. Most important parameters are from $k - l$ strategy. In order to evaluate the influence of its parameters k and l , we conducted a series of experiments. The weights for all the masses are set as the same values.

Fig. 7 illustrates the sensitivity of the $k - l$ selection strategy. By setting the value of k bigger than 10, the $k - l$ strategy ensures that CRET can achieve a high recognition rate. Meanwhile, CRET is capable of tuning the value of the parameter l to control the accuracy degree that it aims to achieve. By tuning the parameters k and l , CRET can infer contexts in an efficient way. In our experiments, we set the values for these two parameters as 10 and 40 in our experiments.

6. Conclusion

Context reasoning is a challenging issue in pervasive computing. One major reason is imprecise and incomplete contexts collected from environments. To this end, we propose CRET – Context Reasoning using Extended Evidence theory, which proves it is possible to assist users in daily life by automatically capturing their contexts and then adapting system behaviors.

Currently, CRET still suffers from several limitations. One limitation is the lack of semantics in context representation. We have to look for relationships among various elements in pervasive computing environments, such as sensors, objects and activities. Ontology-based models are much better in such knowledge representation than both Bayesian network and evidence theory. We are investigating ontology-based models as a feed-in to the propose approach. Another limitation is the central control. We need to implement CRET in a distributed manner, because central control may not be guaranteed in pervasive computing environments that are distributed with users' frequent movement at their will. Finally, we will study how to detect and interpret

concurrent activities in asynchronous and dynamic pervasive computing environments.

Acknowledgments

We would like to thank Vaskar Raychoudhury for his technical discussions. We also thank Nicole Kwoh for her proofreading.

This work is supported by the National High Technology Research and Development Program (863 Program) of China (Grant No. 2006AA01Z172 and 2008AA01Z106), the National Natural Science Foundation of China (Grant No. 60533040, 60725208 and 60773089), and Shanghai Pujiang Program (Grant No. 07pj14049).

This work is also partially supported by Hong Kong Polytechnic Univ under the grant G-U513.

References

- [1] M. Weiser, The computer for the twenty-first century, *Scientific American* 265 (3) (1991) 94–104.
- [2] M. Satyanarayanan, Pervasive computing: vision and challenges, *Personal Communications, IEEE* 8 (4) (2001) 10–17. (See also *IEEE Wireless Communications*).
- [3] C. Xu, S.C. Cheung, Inconsistency detection and resolution for context-aware middleware support, in: *ESEC/FSE-13: Proceedings of the 10th European Software Engineering Conference*, 2005, pp. 336–345.
- [4] X. Chang, S.C. Cheung, W.K. Chan, Y. Chunyang, Heuristics-based strategies for resolving context inconsistencies in pervasive computing applications, in: *ICDCS'08, 2008*, pp. 713–721.
- [5] S. Jeffery, M. Garofalakis, M. Franklin, Adaptive cleaning for RFID data streams, in: *Proceedings of the 32nd International Conference on Very large data bases*, 2006, pp. 163–174.
- [6] E. Elnahrawy, B. Nath, Cleaning and querying noisy sensors, in: *Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications*, 2003, pp. 78–87.
- [7] K. Liu, L. Chen, Y. Liu, M. Li, Robust and efficient aggregate query processing in wireless sensor networks, *Mobile Networks and Applications* 13 (1) (2008) 212–227.
- [8] A. Ranganathan, J. Al-Muhtadi, R.H. Campbell, Reasoning about uncertain contexts in pervasive computing environments, *IEEE Pervasive Computing* 3 (2) (2004) 62–70.
- [9] M. Mamei, R. Nagpal, Macro programming through bayesian networks: Distributed inference and anomaly detection, in: *Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications*, 2007, 2007, pp. 87–96.
- [10] A. Kofod Petersen, M. Mikalsen, Context: Representation and reasoning: Representing and reasoning about context in a mobile environment, *Revue d'intelligence artificielle* 19 (3) (2005) 479–498.
- [11] A. Ranganathan, R.H. Campbell, An infrastructure for context-awareness based on first order logic, *Personal Ubiquitous Comput.* 7 (6) (2003) 353–364.
- [12] J. Shen, Y. Yang, Extending RDF in distributed knowledge-intensive applications, *Future Generation Computer Systems* 20 (1) (2004) 27–46.
- [13] X. Wang, D. Zhang, T. Gu, H. Pung, Ontology based context modeling and reasoning using OWL, in: *Pervasive Computing and Communications Workshops. Proceedings of the 2nd IEEE Annual Conference on*, 2004, pp. 18–22.
- [14] Y.J. Joung, F.Y. Chuang, OntoZilla: An ontology-based, semi-structured and evolutionary p2p network for information systems and services, *Future Generation Computer Systems* 25 (1) (2009) 53–63.
- [15] A. Bikakis, T. Patkos, G. Antoniou, D. Plexousakis, A survey of semantics-based approaches for context reasoning in ambient intelligence, in: *Proceedings of the Workshop Artificial Intelligence Methods for Ambient Intelligence*, Springer, 2007, pp. 15–24.
- [16] G. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, 1976.
- [17] L. Liu, R.R. Yager, Classic works of the Dempster-Shafer theory of belief functions: An introduction, in: *Studies in Fuzziness and Soft Computing*, Vol. 219/2008, Springer, 2008, pp. 1–34.
- [18] P. Orponen, Dempster's rule of combination is # p-complete, *Artificial Intelligence* (1990) 243–245.
- [19] F. Voorbraak, A computationally efficient approximation of Dempster-Shafer theory, *International Journal of Man-Machine Studies* 30 (5) (1989) 525–536.
- [20] L. Zadeh, Review of a mathematical theory of evidence, *AI Magazine* 5 (3) (1984) 81.
- [21] J. Cassens, A. Kofod-Petersen, Designing explanation aware systems: The quest for explanation patterns, in: *Proceedings of the 22nd International Conference on Artificial Intelligence Workshop*, 2007, pp. 303–316.
- [22] J.A.F. Nieto, M.E.B. Gutierrez, B.P. Lancho, Developing home care intelligent environments: From theory to practice, in: *Proceedings of the 7th International Conference on Practical Applications of Agents and Multi-Agent Systems*, 2009, pp. 2–12.
- [23] M. Hedges, T. Blanke, A. Hasan, Rule-based curation and preservation of data: A data grid approach using iRODS, *Future Generation Computer Systems* 25 (4) (2009) 446–452.
- [24] T. Gu, H.K. Pung, D.Q. Zhang, Toward an OSGi-based infrastructure for context-aware applications, *IEEE Pervasive Computing* 3 (4) (2004) 66–74.
- [25] A.D. Sarma, O. Benjelloun, A. Halevy, J. Widom, Working models for uncertain data, in: *Proceedings of the 22nd International Conference on Data Engineering, ICDE'06, 2006*, pp. 162–174.
- [26] P. Korpipaa, J. Mantyjarvi, J. Kela, H. Keranen, E. Malm, Managing context information in mobile devices, *IEEE Pervasive Computing* 2 (3) (2003) 42–51.
- [27] G. Biegel, V. Cahill, A framework for developing mobile, context-aware applications, in: *Pervasive Computing and Communications, Proceedings of the 2nd IEEE Annual Conference on*, 2004, 361–365.
- [28] Y. Ma, D.V. Kalashnikov, S. Mehrotra, Toward managing uncertain spatial information for situational awareness applications, *IEEE Transactions on Knowledge and Data Engineering* 20 (10) (2008) 1408–1423.
- [29] G.F. Cooper, The computational complexity of probabilistic inference using bayesian belief networks (research note), *Artificial Intelligence* 42 (2-3) (1990) 393–405.
- [30] N. Friedman, D. Geiger, M. Goldszmidt, Bayesian network classifiers, *Machine Learning* 29 (2-3) (1997) 131–163.
- [31] S. McClean, B. Scotney, M. Shapcott, Aggregation of imprecise and uncertain information in databases, *IEEE Transactions on Knowledge and Data Engineering* 13 (6) (2001) 902–912.
- [32] R. Murphy, Dempster-Shafer theory for sensor fusion in autonomous mobile robots, *IEEE Transactions on Robotics and Automation* 14 (2) (1998) 197–206.
- [33] X. Hong, C. Nugent, M. Mulvenna, S. McClean, B. Scotney, S. Devlin, Evidential fusion of sensor data for activity recognition in smart homes, *Pervasive and Mobile Computing* 5 (3) (2009) 236–252.
- [34] H. Wu, Sensor data fusion for context-aware computing using DS theory, Ph.D. thesis, Carnegie Mellon University, 2003.
- [35] D. Dubois, H. Prade, Consonant approximation of belief functions, *International Journal of Approximate Reasoning* 4 (5-6) (1990) 419–449.
- [36] R. Yager, On the Dempster-Shafer framework and new combination rules, *Information Sciences: an International Journal* 41 (2) (1987) 93–137.
- [37] P. Smets, The combination of evidence in the transferable belief model, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12 (5) (1990) 447–458.
- [38] A.P. Dempster, A generalization of bayesian inference, *Journal of the Royal Statistical Society, Series B* 30 (1968) 205–247.
- [39] D.Q. Zhang, J.N. Cao, J.Y. Zhou, M.Y. Guo, Extended Dempster-Shafer theory in context reasoning for ubiquitous computing environments, in: *Proceedings of the 7th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, 2009.
- [40] K. Huang, L. Wang, D. Zhang, Y. Liu, Optimizing the BitTorrent performance using an adaptive peer selection strategy, *Future Generation Computer Systems* 24 (7) (2008) 621–630.
- [41] Department of Architecture, Massachusetts Institute of Technology, House_n project. http://architecture.mit.edu/house_n/intro.html.
- [42] N.L. Zhang, D. Poole, Exploiting causal independencies in Bayesian network inference, *Journal of Artificial Intelligence Research* (1996) 301–328.

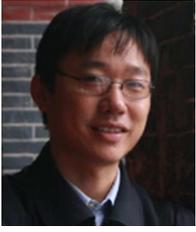


Daqiang Zhang received the B.S. and M.S. degrees from Anhui University in Management and Computer Science in 2003 and 2006, respectively. He is a joint Ph.D. candidate at Shanghai Jiao Tong University and The Hong Kong Polytechnic University, China. His current research focuses on context-awareness in mobile computing, pervasive computing and wireless sensor networks, involving context collection, storage, dissemination, reasoning and privacy and security.



Minyi Guo received the B.S. and M.E. degrees in Computer Science from Nanjing University, China in 1982 and 1986, respectively. He received the Ph.D. degree in Information Science from University of Tsukuba, Japan in 1998. From 1998 to 2000, Dr. Guo had been a research associate of NEC Soft, Ltd. Japan. He was a visiting professor of the Department of Computer Science, Georgia Institute of Technology. He was a full professor at The University of Aizu, Japan and is a full professor at Shanghai Jiao Tong University, China.

He is an IEEE senior member and has published more than 150 papers in well-known conferences and journals. His main interests include automatic parallelization and data-parallel languages, bioinformatics, compiler optimization, high performance computing, and pervasive computing.



Jingyu Zhou received the B.S. degree in Computer Science from Zhejiang University, China, in 1999. He received the M.S. and Ph.D. degrees in Computer Science from University of California at Santa Barbara in 2003 and 2006. He joined the Software Institute at Shanghai Jiao Tong University in 2006.

He is generally interested in information retrieval, systems, and security. His current work is on vertical web search, including information retrieval, Chinese analysis, and search systems. His past work includes: network and application security, parallel scientific computing,

cluster-based storage systems and middleware systems, and cluster-based Internet services.



Dazhou Kang received his BE, ME and Ph.D. degree in computer science from Southeast University at Nanjing, China in 2003, 2005 and 2008. From 2008 to 2009, he is a research associate at The Hong Kong Polytechnic University. Currently, Dr. Kang is a postdoctoral fellow in the Department of Computer Science and Engineering, Nanjing University. His current interests include semantic web, knowledge reasoning and software engineering.



Jiannong Cao is a full professor and associate head of the Department of Computing at Hong Kong Polytechnic University. He is also the director of the Internet and Mobile Computing Lab <http://www4.comp.polyu.edu.hk/%7Eimc/> in the department. Before joined the Hong Kong Polytechnic University in 1997, he has been on faculty of computer science in James Cook University <http://www.jcu.edu.au> and The University of Adelaide <http://www.adelaide.edu.au> in Australia, and the City University of Hong Kong <http://www.cityu.edu.hk>. Jiannong Cao is an adjunct professor of Beijing Jiaotong University <http://www.njtu.edu.cn/en/>, National University of Defense Technology <http://www.nudt.edu.cn/> and Northwest Polytechnic University <http://www.nwpu.edu.cn/>, and a guest professor of Central South University <http://www.csu.edu.cn/english/overview/index.htm> in China. He also held several visiting positions, including a visiting research professor in the National Key Lab for Novel Software Technology <http://keysoftlab.nju.edu.cn/site/ndjsjx/>, Nanjing University of China, a visiting fellow in the School of Computer Engineering <http://www3.ntu.edu.sg/SCE/>, Nanyang Technological University of Singapore, a visiting scholar at the Institute of Software <http://www.ios.ac.cn/english/index.action>, Chinese Academy of Science, and Peking University http://net.pku.edu.cn/Overseas_Scholar_Lecture_Program.

Jiannong. Cao received the B.Sc. degree in computer science from Nanjing University <http://www.nju.edu.cn>, China, in 1982, and the M.Sc. and Ph.D. degrees from Washington State University <http://www.wsu.edu>, USA, in 1986 and 1990, all in computer science. From 1982 to 1983, he studied as a MSc. student in the Graduate School of the Chinese Academy of Science <http://www.gscas.ac.cn/> in Beijing.