A BAYESIAN APPROACH FOR DEALING WITH UNCERTAIN CONTEXTS

Tao Gu[1], Hung Keng Pung[2], Da Qing Zhang[3]

Abstract

Building of context-aware applications in pervasive computing environments faces the difficult problem of dealing with uncertain context information. In this paper, we propose a probability extension to our ontology-based model for representing uncertain contexts; and use Bayesian networks to reason about uncertainty. In addition, the supports of probabilistic markups and Bayesian networks have been incorporated into our context-aware middleware system to enable the building of context-aware services by using various uncertain contexts. We also present our experiences and discussions.

1. Introduction

Emerging pervasive computing technologies provide "anytime, anywhere" computing by decoupling users from devices and viewing applications as entities that perform tasks on behalf of users [1]. To enable this vision, context-awareness is often touted as a key enabler to exhibit the required levels of autonomy and flexibility. Context refers to any information that can be used to characterize the situation of a person, place, or physical or computational object [2]. There have been many interests in making applications and services context aware so that they can exploit various contexts such as user location, profiles and activities, and automatically adapt their behavior in response to dynamic environments and user requirements.

Most of context-aware services and applications assume that the context information upon which they rely is perfect and accurate. However, this assumption is unjustified. In many cases, it may not able to identify context precisely as a result of the limitations of sensing technologies; and hence, high-level contexts which are derived from these inaccurate sensor data may not be accurate. For example, it may be difficult to precisely sense the current location of a user or accurately determine the user activity in a smart home environment. Therefore, how to handle such uncertain contexts is a challenge that many researchers face.

In this paper, we propose a common model for representing uncertain contexts and use Bayesian networks to reason about uncertainty. This model extends our basic ontology-based model by attaching probability values to context predicates. To incorporate probability into context ontologies, we propose a probabilistic extension to OWL [3] - ontology markup language to allow additional probabilistic markups. Bayesian network has become an established probabilistic framework for knowledge representation and inference under uncertainty. We adopt Bayesian network as underlying reasoning mechanism as it has efficient probabilistic reasoning capabilities and allows us to represent causal relationships between various contexts. We incorporate the Bayesian

[1] National University of Singapore, Institute of Infocomm Research, gutao@comp.nus.edu.sg
[2] National University of Singapore, punghk@comp.nus.edu.sg
[3] Institute of Infocomm Research, daqing@i2r.a-star.edu.sg

network inference mechanism into SOCAM [4] which is a context-aware middleware system which provides support for most of the tasks involved in dealing with context - acquiring context from various sources; sharing and interpreting context; and carrying out dissemination of context.

The rest of this paper is organized as follows. Section 2 discusses on related work; in Section 3 we present our model for uncertain contexts. The middleware support for Bayesian networks and its implementation is presented in Section 4. Section 5 presents our evaluation results. Finally, we conclude in Section 6.

2. Related Work

A number of research has been done in addressing uncertain contexts. Dey et al. [5] suggest a mediation process for resolving imperfect sensed contexts. However, this approach may place additional burden on the user as the user is involved in the mediation process. Judd and Steenkiste [6] introduce a contextual information service for querying contexts which allows dynamic contexts to be associated with meta-attributes such as accuracy, confidence, update time and sample interval. Lei et al. [7] propose a context service that allows expressing quality of context information (QoI) such as freshness, confidence and error in context sources and pass along QoI data from sources to clients. Gray and Salber [8] include quality attribute in their context model such as coverage, resolution, accuracy, repeatability, frequency and timeliness. These solutions have several drawbacks. Almost all existing models for uncertain contexts lack of expressiveness to capture rich types of contexts; and they do not support reasoning about various contexts.

3. A Model for Uncertain Contexts

In this section, we discuss our basic context model and its probabilistic extension; and translation from a RDF graph to a Bayesian network.

3.1. A Ontology-based Model

The initial concept for modeling context information has been introduced in [9]. In our model, contexts are represented as first-order predicate calculus. The basic model has the form of *Predicate(subject, value)*, in which

- *subject* $\in$ *S*:* set of subject names, e.g., a person, a location or an object.
- *Predicate* $\in$ *V*:* set of predicate names, e.g., is located in, has status, etc.
- *value* $\in$ *O*:* set of all values of subjects in S*, e.g., the living room, open, close, empty, etc.

For example, *Location(John, bathroom)* - John is located in the bathroom, *Temperature(kitchen, 60)* - the temperature of the kitchen is 60ºC, *Status(door, open)* - the door's status is open, etc.

The structures and properties of context predicates are described in an ontology which may include descriptions of classes, properties and their instances. The ontology is written in OWL as a collection of RDF triples, each statement being in the form (*subject, verb, object*), where subject and object are ontology's objects or individuals, and predicate is a property relation defined by the ontology.

3.2. Probabilistic Extension to the Basic Model

For representing uncertain contexts, we extend our basic context model by incorporating probabilistic information. It has the form of *Prob(Predicate(subject, value)*, in which the probability measurement takes a value between 0 and 1. The extended model applies to any type of contexts such as sensed contexts, defined contexts and derived contexts. For example, in the case of derived context, *Prob(Status(John,Sleeping))= 0.8* means the probability that John is currently sleeping is 0.8.

3.3. Probability-Annotated Ontology

As the ontology language OWL does not provide any support for probabilistic information, we need to augment its capability to allow additional probabilistic markups. To encode probability into a context ontology, we define two OWL classes: "*PriorProb*" and "*CondProb*". Both classes have an object property - "*hasVariable*" and a datatype property - "*hasProbValue*"; and class "*CondProb*" has an additional object property - "*hasCond*". These definitions enable us to specify arbitrary probability. For example, assuming *A, B, C* represent context predicates in the form of RDF triples, then "*P(A)*" - a prior probability, is defined as an instance of class "*PriorProb*" as shown in Figure 1a; and $P(A|B,\overline{C})$ - a conditional probability, is defined as an instance of class "*CondProb*" as shown in Figure 1b.

P (A) = 0.7

```
<prob:PriorProb rdf:ID="P(A)">
  <prob:hasVariable><rdf:value>A</rdf:value></prob:hasVariable>
  <prob:hasProbValue>0.7</prob:hasProbValue>
</prob:PriorProb>
```

(a)

$P(A \mid B, \overline{C}) = 0.5$

```
<prob:CondProb rdf:ID="P(A|B, notC)">
  <prob:hasCond><rdf:value>B</rdf:value></prob:hasCond>
  <prob:hasCond><rdf:value>notC</rdf:value></prob:hasCond>
  <prob:hasVariable><rdf:value>A</rdf:value></prob:hasVariable>
  <prob:hasProbValue>0.5</prob:hasProbValue>
</prob:CondProb>
```

(b)

Figure 1. OWL expressions with probabilistic markups

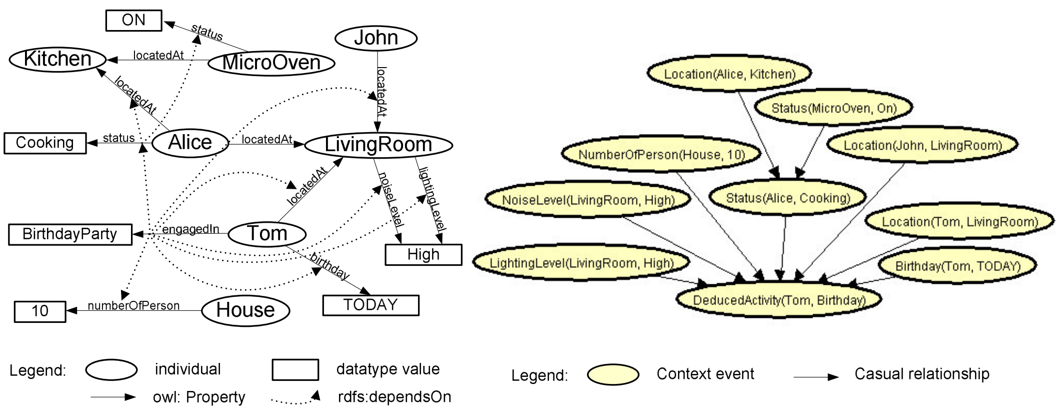
This proposed encoding scheme is influenced by the work in [10], but our scheme allows to markup arbitrary conditional probability such as the example in Figure 1b.

3.4. Bayesian Networks and Structural Translation

A Bayesian network (BN) is a directed acyclic graph (DAG), where each node corresponds a random variable *Xi* and directed arcs represent influential relationships among the random variables. The uncertainty of the causal relationship is represented locally by the conditional probability table *P(Xi|Pa(Xi))* associated with each node *Xi*, where *Pa(Xi)* is the parent set of *Xi*. Under a conditional independence assumption, the joint probability distribution of *X = (X1, . . . ,Xn)* can be obtained as following: $P(X = x) = \prod_{i=1}^{n} P(Xi \mid Pa(Xi))$.

A BN is a powerful graphical tool for representing, learning and computing probability distributions. We apply BN to enable learning casual dependencies between various context events, and obtaining probability distributions. In our model, each node take corresponds to a context predicate; and directed arcs between nodes represent causal relationships between the contexts. By giving the conditional probability table, we are able to compute the probability distribution of a BN.

Constructing a Bayesian network for context information involves identifying causal dependencies between different context events, and translating a context ontology to a BN. Dependency relation exists among various types of context information. A dependency captures the existence of a reliance of property associated with one entity on another. To markup dependency information in an ontology using OWL, we introduce an additional property elements - *rdfs:dependsOn* which captures the dependency relationship of properties associated with datatypes and objects. For example, in Figure 2a, Alice's status (Cooking) may depend on her location (Kitchen), the MicroOven's status (On), etc. As the DAG of a BN and the RDF graph of a context ontology share a structural similarity: both of them are directed graphs, and direct correspondence exists between many nodes and arcs, we are able to translate a RDF graph to a DAG. For structural translation, each context predicate (specified as RDF triples in an OWL file) is mapped into a node in the BN, and an arc is drawn between two nodes if and only if there exists a dependency relation between two context predicates. The example in Figure 2 shows a RDF graph augmented with dependency markup is translated into a DAG. The derived context - Tom's current activity, depends on his location, the living room's lighting level and noise level, his parent's status and location, number of person in his house, his profile (Birthday), etc.



(a) A partial RDF graph of context ontologies (b) A DAG of Bayesian network

Figure 2. Translate a RDF graph to a DAG

4. Middleware Support for Using Bayesian Networks

In this section, we describe our context-aware middleware system which provides supports to construct a BN and reason about uncertain contexts.

4.1. The SOCAM Middleware

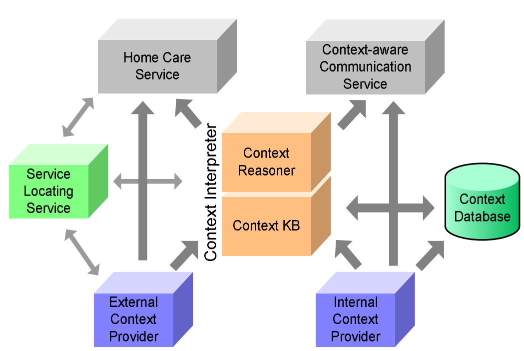


Figure 3. Overview of the SOCAM architecture

Based on our context model, we design a Service-Oriented Context-Aware Middleware (SOCAM) architecture which aims to enable building and rapid prototyping of context-aware services. It consists of the following components as shown in Figure 3.

- Context Providers: They abstract useful contexts from heterogeneous sources - external or internal; and convert them to OWL expressions so that contexts can be shared and reused by other entities.
- Context Interpreter: It provides logic reasoning services including inferring indirect contexts from direct contexts, querying context knowledge, etc. It consists of context reasoning engines and a context KB (Knowledge Base).
- Context Database: It stores context ontologies and past contexts for a sub-domain. There is one logic context database in each domain, i.e., home domain.
- Context-aware Services: They access and use different level of contexts and adapt the way they behave according to the current context.
- Service Locating Service [11]: It provides a mechanism where context providers and the context interpreter can advertise their presence; also enables users or applications to locate and access these services.

4.2. Reasoning about Uncertain Contexts

In SOCAM, different context providers register and advertise their services through the Service Locating Service. Context consumers, i.e., the context interpreter or context-aware services, are able to locate a context provider and obtain a context of interest. Context dissemination is done in both push and pull modes. Users or services can either issue a query for a particular piece of context or subscribe a context event to a context provider. When the event is triggered, the particular context in the form of OWL expressions will be returned to the subscriber.

The supports for using probability-annotated context ontology and BN have been built in SOCAM. The context ontology with additional dependency markups is created and stored in a context database. SOCAM can translate this ontology into a BN based on the structural translation rule described in Section 3.4. After the BN is created, it is trained on real data to get probability distributions for the various nodes. SOCAM provides supports for getting data from the environment and training the BN. All the past contexts are logged in a database; and the conditional probability table (CPT) to each node can be computed. Once the BN is trained, it is used for inferring the probabilities of context conditions and other events. We have built-in the Bayesian network software toolkit - BNJ [12] into the context interpreter as the underlying reasoning mechanism. Reasoning on uncertainty in BN takes the assigning probability values to a set of nodes and then propagating the influence of these assignments to other nodes in the network. By training a set of data, we are able to obtain the probabilities of all root nodes (node with no predecessors) and the conditional probabilities of all non-root modes. Hence, the probability distributions of various context events in the BN can be found. These probabilities in the form of probability-annotated OWL expressions will be added to the context knowledge base for query and access. To use uncertain contexts, service developers will need to specify actions that are triggered by a set of pre-defined rules. These rules typically include uncertain contexts and a set of conditions.

5. Evaluation

The experiment is to infer a person's current activity (i.e., birthday party, reading, watching TV, dining, sleeping, etc) in a smart home environment. First, a context ontology with dependency markups was created and stored in the context database. The dependency relationship was defined in a form of user-specified rules such as *DeducedActivity(User, X): Location(User, Y), LightingLevel(Y, ?x), NoiseLevel(Y, ?x), NumberOfPerson(Y, ?x), Profile(User, ?x), Location(OtherMember, ?x).* Based on the ontology, the context interpreter was able to generate a Bayesian network. The context interpreter got all training data and assigned probability distributions to each node. For example,

<prob:PriorProb rdf:ID="P(Tom locatedIn LivingRoom)">

<prob:hasVariable><rdf:value>(Tom locatedIn LivingRoom)</rdf:value><prob:hasVariable>

<prob:hasProbValue>0.91</prob:hasProbValue>

</prob:PriorProb>

The interpreter performed the Bayesian reasoning to compute the joint probability. In our test, we took 336 observations for a period of 2 weeks. The interpreter performed fairly well and was able to assign the right activity with the highest probability in most of the cases.

Our experiences show that Bayesian network is a powerful method of reasoning about causal relationships between various uncertain contexts. It is flexible and can be retrained easily. The limitation of a Bayesian approach for handling uncertainty in pervasive computing environments is that it may be difficult to get data to train a BN in certain circumstances such as in the application of security control.

6. Conclusions

In conclusion, we propose a probability model for uncertainty in pervasive computing environments and use Bayesian networks to reason about such uncertainty. This approach can make context-aware applications more robust and more capable of adapting to the changing environment. We are also looking at other efficient methods to reason about uncertain contexts. For example, fuzzy logic may be useful to represent and reason about imprecise notions such as "hot", "very low", "confidence", etc.

7. References

[1] Henricksen K, Indulska J, Rakotonirainy, "Infrastructure for Pervasive computing: Challenges", Workshop on Pervasive Computing INFORMATIK 01, Viena, 2001.

[2] Dey, A. and Abowd, G., "Towards a Better Understanding of Context and Context-Awareness", Workshop on the what, who, where, when and how of context-awareness at CHI 2000, April 2000.

[3] OWL Web Ontology Language Overview, http://www.w3.org/TR/owl-features.

[4] T. Gu, H. K. Pung, D. Q. Zhang, "A Middleware for Building Context-Aware Mobile Services", In Proceedings of IEEE Vehicular Technology Conference, Milan, Italy, May 2004.

[5] Dey, A., Manko., J., Abowd, G. "Distributed mediation of imperfectly sensed context in aware environments", Technical Report GIT-GVU-00-14, Georgia Institute of Technology, 2000.

[6] G. Judd, P. Steenkiste, "Providing Contextual Information to Pervasive Computing Applications", In Proceedings of the 1st IEEE Conference on Pervasive Computing and Communications, Fort Worth, Texas, March 2003.

[7] Hui Lei, Daby M. Sow, John S. Davis, II, Guruduth Banavar and Maria R. Ebling, "The design and applications of a context service", ACM SIGMOBILE Mobile Computing and Communications Review, vol 6, no. 4, pp 44-55, 2002.

[8] Gray, P., Salber, D. "Modeling and using sensed context in the design of interactive applications", In Proceedings of 8th IFIP Conference on Engineering for Human-Computer Interaction, Toronto, 2001.

[9] T. Gu, X. H. Wang, H. K. Pung, D. Q. Zhang, "An Ontology-based Context Model in Intelligent Environments", In Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference, San Diego, California, January 2004.
[10] Zhongli Ding, Yun Peng. "A Probabilistic Extension to Ontology Language OWL", In Proceedings of the 37th Hawaii International Conference on System Sciences, Hawaii, January 2004.
[11] T. Gu, H. C. Qian, J. K. Yao, H. K. Pung, "An Architecture for Flexible Service Discovery in OCTOPUS", In Proceedings of the 12th International Conference on Computer Communications and Networks, Dallas, Texas, October 2003.
[12] Bayesian Network tools in Java (BNJ), http://bndev.sourceforge.net/