# Using Fuzzy Decision Tree to Handle Uncertainty in Context Deduction

Donghai Guan, Weiwei Yuan, A. Gavrilov, Sungyoung Lee*, Youngkoo Lee and Sangman Han

Department of Computer Engineering
Kyung Hee University, Korea
{donghai,weiwei,avg,sylee,i30000}@oslab.khu.ac.kr yklee@khu.ac.kr

**Abstract.** In context-aware systems, one of the main challenges is how to tackle context uncertainty well, since perceived context always yields uncertainty and ambiguity with consequential effect on the performance of context-aware systems. We argue that uncertainty is mainly generated by two sources. One is sensor's inherent inaccuracy and unreliability. The other source is deduction process from low-level context to high-level context. Decision tree is an appropriate candidate for reasoning. Its distinct merit is that once a decision tree has been constructed, it is simple to convert it into a set of human-understandable rules. So human can easily improve these rules. However, one inherent disadvantage of decision tree is that the use of crisp points makes the decision trees sensitive to noise. To overcome this problem, we propose an alternative method, fuzzy decision tree, based on fuzzy set theory.

## 1 Introduction

Since first been proposed by Weiser in the early 1990s, ubiquitous computing has been one of the predominant trends in computing over last ten years. In a ubiquitous computing environment, computers will be everywhere around us without our awareness. In other words, computers will have moved into background.

Usually, ubiquitous system makes intelligent decisions by analyzing context information. Context refers to any information that can be used to characterize the situation of an entity. Here, an entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves [1].

Context is characterized at different levels of abstraction: low-level and high-level. Low-level context (such as temperature, light, voice level) is gathered directly from physical sensors. While high-level context is abstract and inferred from low-level context. For example, User's activity is a kind of high-level context. It can be inferred through some low-level context.

* Prof. Sungyoung Lee is the corresponding author.

Context is important for system to sense, in turn, think and act. However, one potential problem for context is that it is uncertain. The ability to handle context uncertainty has become one of the main challenges in context-aware computing [2] [3] [4] [5].

Context uncertainty comes from many sources. Firstly, sensors are usually not fully reliable. For instance, a motion detector may not be able to detect people in one hundred percent of cases. On the other hand, some sensors may be more prone to cause false alarms, e.g., a face detector claims it has recognized a particular person while it has not [6]. Inherent inaccuracy and unreliability of many sensors makes low-level context uncertain. High-level context uncertainty comes from deduction process itself. Any deduction (inferring) process is uncertain.

Some methods have been proposed to deal with high-level context uncertainty. Bayesian networks is supposed to handle this problem well [6], [7], especially when there are causal relationships between various events. Also, probabilistic logic and fuzzy logic are used to handle uncertainty in [7]. Above methods can solve uncertainty to some extent. However, we argue that all of them are not perfect. The main function of ubiquitous system is to read users' mind and provide appropriate service to them. One distinct requirement is that user need to understand system's reasoning process so that if system acts unreasonably, user can correct it. Another requirement is reasoning algorithm should be powerful enough so that it still can work well in some complicated cases.

All of the above methods cannot fully meet the two requirements. As for Bayesian networks, although its reasoning ability is powerful, it cannot be easily converted into rules. As for probabilistic and fuzzy logic, their reasoning process is expressed by a set of rules. However, it is hard for users to make rules for complicated situations. Decision tree seems to be an appropriate choice. It can be simply to convert into a set of rules. Also, it works well even in some complicated cases. However, one inherent disadvantage is the use of crisp cut points makes the induced decision tree sensitive to noise. To overcome this problem, we propose an alternative method, called fuzzy decision tree, based on fuzzy set theory. This method has been successfully applied to an industrial problem to monitor a typical machining process.

## 2 High-level Context Reasoning

High-level context is derived through low-level context fusion, aggregation or generalization. The advantage of high-level context is that it provides more explicit and useful result for application, which is always implicit from the point of low-level context. High-level context is more effective when predicting user's need and delivering appropriate service to user. In a smart office scenario, "five persons in room now" and "projector is working" are low-level context. And from them, we may deduce a high-level context—"they are having meeting now".

Many machine learning techniques are able to achieve this kind of reasoning task. Such as decision tree, neural network, Bayesian networks. Here, we choose decision tree, not for its more powerful reasoning ability than others, but for its result is easily transformed to rules. This is important because users can directly see the rules and

they also can change the rules if these deduced rules are explicitly unreasonable. However, if we select neural network or Bayesian networks, the results are not readable. Sometimes the system's prediction will confuse us and the worst thing is that we have no any idea why system does like this and how to solve it.

To clearly illustrate our method, we devise a scenario. In this scenario, the ubiquitous computing environment is a smart office. The low-level context includes: time, temperature, humidity, light and so on, which could be directly got from sensors. The high-level context is "deducing whether or not some specified devices should be automated selected and work". To easily understand this scenario, the devices here are only referred to heater and humidifier.

Although decision tree learning is able to generate readable results, before using it, we should know whether it is suitable to solve the problem in our scenario. Actually, decision tree learning is generally best suited to problems with the following characteristics [8]:

1) Instances are represented by attribute-value pairs**.**

2) Instances are described by a fixed set of attributes (e.g., temperature) and their values (e.g., hot).

3) The easiest situation for decision tree learning occurs when each attribute takes on a small number of disjoint possible values (e.g., hot, mild, cold).

4) Extensions to the basic algorithm allow handling real-valued attributes as well (e.g., a floating point temperature).

5) The target function has discrete output values**.** A decision tree assigns a classification to each example. Simplest case exists when there are only two possible classes (Boolean classification). Decision tree methods can also be easily extended to learning functions with more than two possible output values.

6) A more substantial extension allows learning target functions with real-valued outputs, although the application of decision trees in this setting is less common.

7) The training data may contain errors. Decision tree learning methods are robust to errors - both errors in classifications of the training examples and errors in the attribute values that describe these examples.

8) The training data may contain missing attribute values. Decision tree methods can be used even when some training examples have unknown values (e.g., humidity is known for only a fraction of the examples).

When we use decision tree method in our devised scenario, the input might include time, temperature, light, humidity or other more context information that can be acquired directly from sensors, and the output is Boolean functions, which is the result whether user will operate on some devices (heater, humidifier).

Here, we just use classical decision tree method, so we should transform real-valued attributes to disjoint values. Our experiment data is shown in table 1.

In table 1, "on" means user turns on the device and "off" means user turns off that device. We have mentioned that the attribute value form should be changed from real to disjoint. The method that transform context from real-valued to disjoint values is shown in table 2.

**Table 1.** Training data for decision tree

| Low-level context | | | | | High-level context | | |
|---|---|---|---|---|---|---|---|
| Time | Temp | Humidity | Light | Others | Heater | Humidifier | Others |
| 9:00 | 25 | 0.2 | 50 | | off | on | |
| 9:05 | 26 | 0.3 | 51 | | off | off | |
| 9:10 | 27 | 0.4 | 52 | | on | on | |
| 9:15 | 28 | 0.5 | 53 | | on | off | |
| 9:20 | 27 | 0.6 | 53 | | off | off | |
| 9:25 | 27 | 0.5 | 52 | | off | on | |
| 9:30 | 26 | 0.4 | 50 | | off | on | |
| 9:35 | 22 | 0.3 | 48 | | on | on | |
| 9:40 | 23 | 0.2 | 49 | | on | on | |
| 9:45 | 22 | 0.2 | 50 | | on | off | |
| 9:50 | 21 | 0.3 | 48 | | off | off | |
| 9:55 | 19 | 0.3 | 49 | | off | on | |
| 10:00 | 18 | 0.5 | 48 | | on | off | |
| 10:05 | 17 | 0.6 | 47 | | on | off | |

**Table 2.** Real value to disjoint value transformation

| Temp | Humidity | Light | |
|---|---|---|---|
| $14 \le T < 20$ | $0.2 \le H < 0.4$ | $30 \le L < 45$ | Low |
| $20 \le T < 25$ | $0.4 \le H < 0.5$ | $45 \le L < 60$ | Middle |
| $25 \le T \le 28$ | $0.5 \le H \le 0.6$ | $60 \le L \le 75$ | High |

Then the transformed training data is shown in table 3.

The first step in building a decision tree is finding the root node. For this purpose, the information gain for each low-level context must be calculated. In the following calculation, S refers to the whole set of training data and the base of the logarithm is 2. The formulas for the calculation of entropy and information gain for "heater" are as follows:

$$E(h) = -\frac{m_{on}}{m_{heater}} log \frac{m_{on}}{m_{heater}} - \frac{m_{off}}{m_{heater}} log \frac{m_{off}}{m_{heater}} \quad (1)$$

Where,

$m_{on}$ is number of tuples, in which Heater= "on"

$m_{off}$ is number of tuples, in which Heater= "off"

$m_{heater} = m_{on} + m_{off}$

**Table 3.** Transformed training data for decision tree

| Low-level context | | | | | High-level context | | |
|---|---|---|---|---|---|---|---|
| Time | Temp | Humidity | Light | Others | Heater | Humidifier | Others |
| 9:00 | Mid | Low | Mid | | off | on | |
| 9:05 | High | Low | Mid | | off | off | |
| 9:10 | High | Mid | Mid | | on | on | |
| 9:15 | High | High | Mid | | on | off | |
| 9:20 | High | High | Mid | | off | off | |
| 9:25 | High | High | Mid | | off | on | |
| 9:30 | High | Mid | Mid | | off | on | |
| 9:35 | Mid | Low | Mid | | on | on | |
| 9:40 | Mid | Low | Mid | | on | on | |
| 9:45 | Mid | Low | Mid | | on | off | |
| 9:50 | Mid | Low | Mid | | off | off | |
| 9:55 | Low | Low | Mid | | off | on | |
| 10:00 | Low | High | Mid | | on | off | |
| 10:05 | Low | High | Mid | | on | off | |

$$G(h,lc) = E(h) - \frac{m_x}{m_{lcm}} E(h, lc = x) - \frac{m_y}{m_{lcm}} E(h, lc = y) - \frac{m_z}{m_{lcm}} E(h, lc = z) \quad \textbf{(2)}$$

Where,

$m_x$ is number of tuples, in which low-level context lcm=x

$m_y$ is number of tuples, in which low-level context lcm=y

$m_z$ is number of tuples, in which low-level context lcm=z

$m_{lcm} = m_x + m_y + m_z$

Therefore, the entropy of the whole set and the information gain for Temp can be calculated as follows:

$$E(h) = -\frac{6}{14}\log\frac{6}{14} - \frac{8}{14}\log\frac{8}{14} = 0.985$$

$$G(h, temp) = E(h) - \frac{m_{high}}{m_{Temp}} E(h, T = high) - \frac{m_{mid}}{m_{Temp}} E(h, T = mid) - \frac{m_{low}}{m_{Temp}} E(h, T = low)$$

$$= 0.985 - \frac{6}{14} E(h, T = high) - \frac{5}{14} E(h, T = mid) - \frac{3}{14} E(h, = low)$$

$$G(h,T) = 0.048 = -\frac{2}{6}\log\frac{2}{6} - \frac{4}{6}\log\frac{4}{6} = 0.918$$

$$E(h,T = mid) = -\frac{3}{5}\log\frac{3}{5} - \frac{2}{5}\log\frac{2}{5} = 0.971$$

$$E(h,T = low) = -\frac{2}{3}\log\frac{2}{3} - \frac{1}{3}\log\frac{1}{3} = 0.918$$

Therefore,

$G(h,T) = 0.048$

Accordingly, the formulas for the calculation of entropy and information gain for "humidifier" are as follows:

$$E(hu) = -\frac{m_{on}}{m_{humi}}\log\frac{m_{on}}{m_{humi}} - \frac{m_{off}}{m_{humi}}\log\frac{m_{off}}{m_{humi}} \quad (3)$$

$$G(hu,lc) = E(hu) - \frac{m_x}{m_{lc}}E(hu,lc = x) - \frac{m_y}{m_{lc}}E(hu,lc = y) - \frac{m_z}{m_{lc}}E(hu,lc = z) \quad (4)$$

Using the formulas above, the information gain of each low-level context could be calculated. Then, the context with highest information gain would be selected as root node of decision tree.

## 3 Fuzzy Logic and Fuzzy Decision Tree

Using decision tree method, we could deduce the high-level context. However, still one main problem exists. For example, if the temperature is $14\,^oC$ or $20\,^oC$, using the method shown in table 2, both of them belong to "Low" category. However, the "Low degree" of 14 and 20 are same? Also $19\,^oC$ and $20\,^oC$ belong to different category based on table 2, so the deduction result might be totally different. The difference between 19 and 20 is only 1. From this example, it is easily to see the method in table 2 is unreasonable.

The main reason is crisp cut points are used in classical decision trees. In fact, crisp cut model does not match human thing and is not reasonable. This makes decision trees sensitive to noise. To overcome this problem, we incorporate fuzzy theory in decision trees. Instead of crisp boundaries between categories, fuzzy logic introduces a membership function, which reflects how well a given value falls into a category. For example, we can define membership for Temperature as follows:

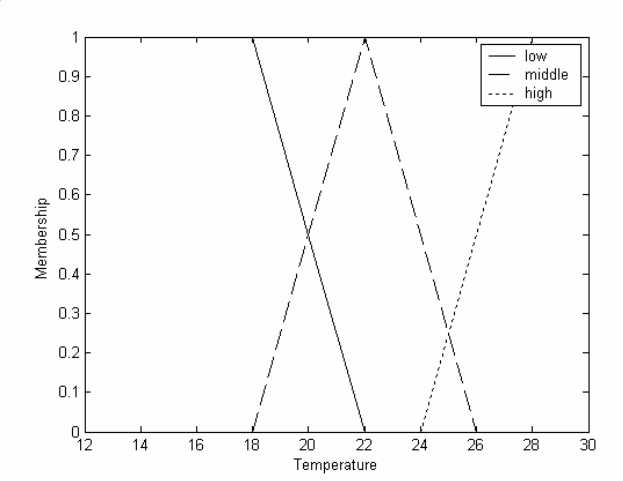


**Fig. 1.** Membership function for temperature

In Fig.1., 0 represents complete non-membership and 1 represents complete membership, while other values representing the degree of membership, or the degree to which the low-context is represented by the linguistic indicator, such as “high”, “middle” and “low”.

Using the example partitions shown in figure 1, the temperature $20\,^{o}C$ would be discretised into the categorical value ‘middle’ with the membership value 0.5, and the temperature $22\,^{o}C$ would be discretised into the same categorical value but with the membership value 1. Consequently, even though the two temperatures $20\,^{o}C$ and $22\,^{o}C$ have the same categorical value ‘middle’, they have different membership values.

Also, other low-level context can be defined using the same method like Figure 1. After all the low-level context have their own membership functions, a corresponding high-level context’s membership functions can be derived using the following formulas:

$$mf_{hc} = mf_{lc1} mf_{lc2} mf_{lc3} ... mf_{lcn} \quad \textbf{(5)}$$

Where,

$mf_{hc}$ refers to membership function of high-level context

$mf_{lcn}$ refers to membership function of low-level context to a given category

For example, if we only consider three low-level context: Temperature, Light, Humidity.

Temperature is middle with membership 0.8

Humidity is low with membership 0.7

Light is middle with membership 0.6

Then, the corresponding high-level context membership is

$mf_{hc} = 0.8 * 0.7 * 0.6 = 0.336$

For illustration, we use the following context information.

**Table 4.** Transformed training data for fuzzy decision tree

| Low-level context | | | | | High-level context | |
|---|---|---|---|---|---|---|
| Item | Temp | Humi | Light | Others | Humidifier | Membership |
| 1 | Mid | Low | Mid | | on | 0.5 |
| 2 | High | Low | Mid | | on | 0.3 |
| 3 | High | Mid | Mid | | on | 0.55 |
| 4 | High | High | Mid | | off | 0.4 |
| 5 | High | High | Mid | | off | 0.3 |
| 6 | High | High | Mid | | on | 0.2 |
| 7 | High | Mid | Mid | | off | 0.25 |
| 8 | Mid | Low | Mid | | on | 0.45 |
| 9 | Mid | Low | Mid | | on | 0.6 |
| 10 | Mid | Low | Mid | | on | 0.6 |
| 11 | Mid | Low | Mid | | off | 0.5 |
| 12 | Low | Low | Mid | | on | 0.7 |
| 13 | Low | High | Mid | | off | 0.45 |
| 14 | Low | High | Mid | | off | 0.5 |

The first step in building a fuzzy decision tree is also finding the root node. So the information gain for each low-level context must be calculated. The formulas for the calculation of entropy and information gain are essential the same as those for building a conventional decision tree. However, in the case of fuzzy decision tree, the membership values of the high-level context should be used in the calculation. The corresponding formulas for the entropy and information gain G are as follows. Here the high-context is humidifier operation prediction.

$$E(hu) = -\frac{m_{on}}{m_{hu}} log \frac{m_{on}}{m_{hu}} - -\frac{m_{off}}{m_{hu}} log \frac{m_{off}}{m_{hu}} \quad (6)$$

Where,

$m_{on} = \sum mf_{humidifier}(on)$ sum of all membership values for humidifier=on

$m_{off} = \sum mf_{humidifier}(off)$ sum of all membership values for humidifier=off

$m_{hu} = m_{on} + m_{off}$

$$G(hu,lc) = E(hu) - -\frac{mf_x}{m_{lc}} E(hu, A = x) - \frac{mf_y}{m_{lc}} E(hu, A = y) - \frac{mf_z}{m_{lc}} E(hu, A = z) \quad (7)$$

Where

$mf_x$ sum of all membership values for A=x

$mf_y$ sum of all membership values for A=y

$mf_z$ sum of all membership values for A=z

$m_{lcm} = mf_x + mf_y + mf_z$

Therefore, the entropy of the whole training set and the information gain for Humidity can be calculated as follows:

$m_{on}$ =0.5+0.3+0.55+0.2+0.45+0.6+0.6+0.7=3.9

$m_{off}$ =0.4+0.3+0.25+0.5+0.45+0.5=2.4

$$E(hu) = -\frac{3.9}{6.3}\log\frac{3.9}{6.3} - \frac{2.4}{6.3}\log\frac{2.4}{6.3} = 0.96$$

$$E(hu, hu = high) = -\frac{1.65}{1.85}\log\frac{1.65}{1.85} - \frac{0.2}{1.85}\log\frac{0.2}{1.85} = 0.49$$

$$E(hu, hu = middle) = -\frac{0.55}{0.8}\log\frac{0.55}{0.8} - \frac{0.25}{0.8}\log\frac{0.25}{0.8} = 0.90$$

$$E(hu, hu = low) = -\frac{3.15}{3.65}\log\frac{3.15}{3.65} - \frac{0.5}{3.65}\log\frac{0.5}{3.65} = 0.58$$

$$G(hu,hu) = E(hu) - -\frac{1.85}{6.3} E(hu, hu = high) - \frac{0.8}{6.3} E(hu, hu = mid) - \frac{3.65}{6.3} E(hu, hu = low)$$

=0.96-0.14-0.11-0.34

=0.37

In the case of attributes Light, Temperature, information gains calculation is same with Humidity.

Then, the context with highest information gain would be selected as root node of fuzzy decision tree. After the root node is found, we can use the similar way to find the other leaf nodes.

## 4 Conclusions

In this paper, firstly, we discuss the main sources of context uncertainty. In addition to sensor's inherent inaccuracy and unreliability, high-level context reasoning is also a main source of uncertainty. Furthermore, we propose to use fuzzy decision tree based algorithm to reason high-level context. Fuzzy decision tree is the extension of classical decision tree by incorporating fuzzy set based approach.

There are two main merits to use this approach. First, uncertainty is reduced so that system reliability is improved. What's more, fuzzy decision tree can be easily converted into human readable rules, which makes it possible for users understand system response and improve system performance by directly changing those rules.

## Acknowledgement

The research was supported by the Driving Force Project for the Next Generation of Gyeonggi Provincial Government in Republic of Korea.

## Reference

1. Dey, A.K., et al.: A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. In J. of Human-Computer Interaction (HCI), Vol. 16. (2001) 97-166
2. Bardram J. E.: Applications of Context-Aware Computing in Hospital Work – Examples and Design Principles. In Proc. of ACM Symposium on Applied Computing (ACM SAC) (2004) 1574-1579
3. Dey A. K., Jennifer Mankoff, Gregory D. Abowd and Scott Carter: Distributed Mediation of Ambiguous Context in Aware Environments. In Proc. of the 15th Annual Symposium on User Interface Software and Technology (UIST 2002), Paris (2002) 121-130
4. Satyanarayanan M.: Pervasive Computing: Vision and Challenges. In IEEE PCM August (2001) 10-17
5. Satyanarayan M,: Coping with Uncertainty. In IEEE CS Pervasive computing Journal, August (2001) 2-3
6. Michal: Data Mining Techniques in Ubiquitous Computing, IWCIT03 (2003)
7. Anand Ranganathan, Jalal Al-Muhtadi, Roy H. Campbell: Reasoning about Uncertain Contexts in Pervasive Computing Environments, In J. IEEE Pervasive Computing, Vol. 3, (2004) 62-70
8. T. Mitchell: Decision Tree Learning, in T. Mitchell, Machine Learning, The McGraw-Hill Companies, Inc. (1997)