Teaching Context to Applications

Kristof Van Laerhoven and Kofi Aidoo

Starlab Research NV/SA, Brussels, Belgium

Abstract: Although mobile devices keep getting smaller and more powerful, their interface with the user is still based on that of the regular desktop computer. This implies that interaction is usually tedious, while interrupting the user is not really desired in ubiquitous computing. We propose adding an array of hardware sensors to the system that, together with machine learning techniques, make the device aware of its context while it is being used. The goal is to make it learn the context-descriptions from its user on the spot, while minimising user-interaction and maximising reliability.

Keywords: Adaptive interfaces; Context awareness; Human-computer interaction; Kohonen self-organising map; On-line adaptive clustering; Sensor data fusion

1. Introduction

1.1. From sensors to contexts

Some applications enhance their user interface by adding a sensor and using the sensors' value in some simple rule. A typical example is connecting a light sensor to a screen-based device and adjusting the contrast and brightness of the screen according to the value of the light sensor.

Other applications can change their behaviour only when the user explicitly tells them to. It is also possible to use user-defined profiles that describe the devices' behaviour. For example, profiles in mobile phones can be set to make the phone ring very loudly outside or on the train but only vibrate in a meeting. This approach leads to a lot of user-involvement, however: the user first needs to program these profiles, and then the profiles must be set in every context ("in a meeting", "in the train"...).

The combination of all of the approaches mentioned earlier leads to an *automated profiles selection*: context recognition based on simple sensors sets the behaviour of the device (see [1] and [2]). Knowing the context usually leads to being able to improve the application and particularly to enhancing the interaction with the user. This approach is far from simple, however: how can a device, equipped with sensors, recognise a context?

1.2. Context

The notion of context is very broad and incorporates lots of information, not just about the current location, but also about the current activity, or even the inner state of the person describing it. As a consequence, people can describe their contexts in different ways, even if they are in the same location doing the same thing. Someone familiar with a building might know a room as "classroom 402B", while a visitor would probably describe it as just "a classroom".

In addition, the application also defines the description of the context. Some applications require more location-based contexts, while others need contexts that give more information about the user. Since contexts depend heavily on both user and application, context awareness should be *adaptive*. Furthermore, to make the device usable the user should be able to give *minimal feedback* to the learning module.

1.3. Context description

The sensors we have experimented with are small, low-level and cheap. The hardware boards (see [3]) that were used (Fig. 1) include light sensors, temperature sensors, accelerometers for movement, microphones, pressure-, IR-, touch- and CO sensors.

The simplest method for giving a context description would be to sum up all the values from the sensors to a formatted description, like for instance "movement: (87%, 29%), light:

Personal and Ubiquitous Computing (2001) 5:46–49

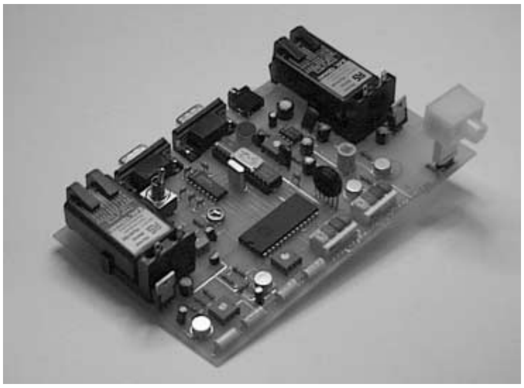

Fig. 1. One of the sensor boards.

78%, humidity: 69%, temperature: 50%...". A simple, rule-based architecture could be used to enhance this description into "moving slowly in a cold, humid, well-lit room".

The architecture described here works the opposite way: the system merges the output from the sensors and maps them to a description given by the user. The description could then be something like "walking in the basement".

2. Online Adaptive Context Awareness

Instead of simply using the raw sensor values as input for the next layer, small pre-processing routines were used to enhance the future clustering. For example, instead of just looking at the brightness of the light, it is also possible to look at its frequency, which results in easier distinguishing of several types of artificial light. Taking the standard deviation of the accelerometer values can also give more qualitative information. Other sensors like microphones and infrared sensors have similar mini-transformations from the raw sensor data to one, usually multiple, value(s), which are often called cues or features.

Another advantage of the cues is that that they are sent less frequently to the next layer. The light sensor, for instance, is read a few hundred times per second. The cues from this sensor (light level and frequency) are sent every second. Cues are very significant for a fast but accurate context recognition system. However, using cues results in a large input dimension, which makes the mapping-algorithm very slow in learning. This difficulty arises when many irrelevant inputs are present and is usually referred to as *the curse of dimensionality* (see [4] for a definition).

2.1. Self-organisation

When a rat has learned its location in a labyrinth, certain braincells on the hippocampal cortex respond only when it is in a particular location. Self-organisation of neuronal functions seems to exist on very abstract levels (like geographic environments) in the brain. The Kohonen Self-Organising Map (SOM) [5] has a similar principle: neurons (artificial, this time) are activated topologically for tasks depending on the sensory input. The SOM is also known to handle noisy data relatively well, which makes it a sensible choice for clustering the inputs.

It is possible to monitor the activation of the neurons and plot the resulting matrix as a landscape, where different hills ideally represent different contexts. This might be a way of providing the user an insight into the learning capabilities of the system (Fig. 2).

The traditional algorithm starts by being highly adaptive (a large learning rate and huge neighbourhood radius) and gradually becomes fixed. After this stage, it is not capable of learning any more, which poses an obstacle if the system needs to remain adaptive. This is a problem also known as *the stability-plasticity dilemma*. Therefore it is necessary to add some supervision mechanism that controls the flexibility of the SOM.

The only necessary user interaction is the labelling of clusters produced by the SOM. This means that when a cluster gets activated, two possible situations can occur: (1) the cluster is

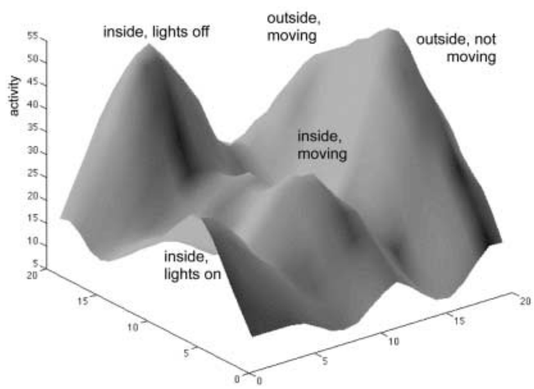

Fig. 2. Example of an activity-plot of a SOM.

labelled, so classification is possible, or (2) there is no label. In the latter case, we use a distance weighted K-Nearest Neighbours algorithm to search for the closest label on the Self-Organising Map. The topology preserving property of the SOM makes it very probable that the nearest label will indeed be the right context.

2.2. Supervision and user behaviour

The next layer is primarily intended to supervise transitions from one context to another. It uses a probabilistic finite state machine architecture where each context is represented by a state, and transitions are represented by edges between states. The model keeps a probability measure for each transition, so every time a transition occurs, the supervision model can check if this really is likely. If a transition is not really probable, the next state is not entered yet, but a buffer mechanism is initiated so that it does become more likely after several tries in a row. Each transition to a state is thus dependent on the previous state, which makes this model a first-order Markov model. Every state also keeps track of how much time was spend in a particular context, which controls the flexibility of the SOMs: the newer a context, the more flexible and adaptive the map should be.

The result is that after some time this model generates a graph depicting the behaviour of a user with relation to the contexts visited. When the user tends to go from A to B rather than to C, then this will be reflected in the graph's connection strengths. Figure 3 depicts the typical layout of the final architecture.

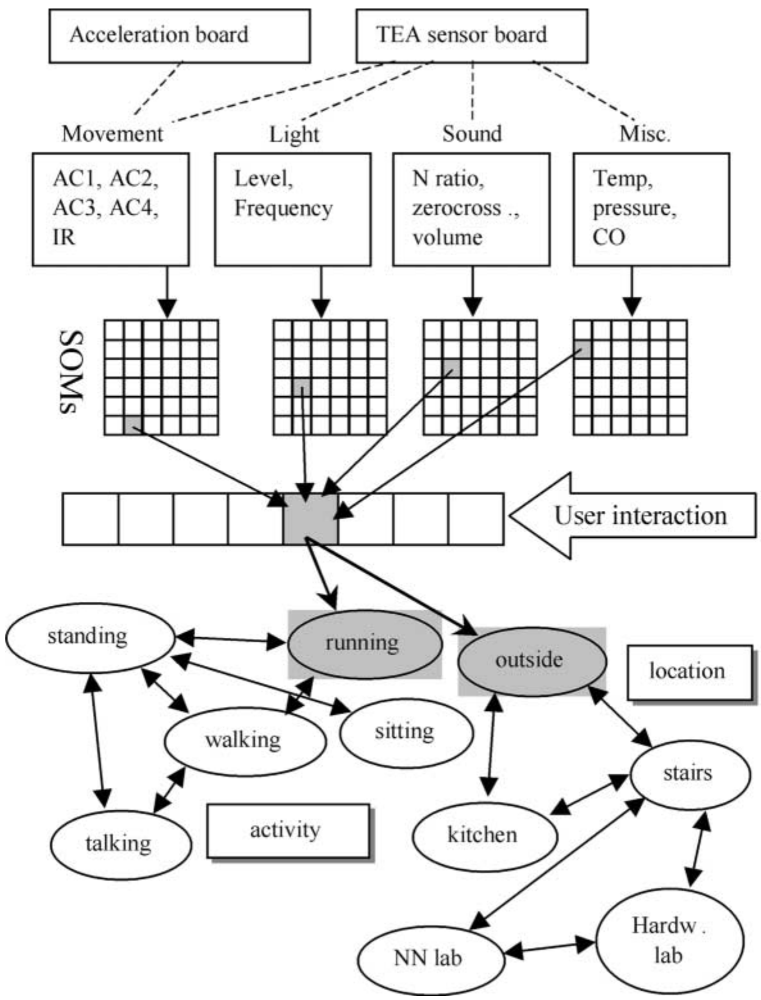

Fig. 3. Overall architecture. User interaction is only necessary after the clustering.

3. Results and Future Work

For context awareness to be effectively user-friendly, it is necessary that the system gets feedback from the user whenever the *user* would like to give it. These constraints are both hard and challenging from a machine-learning point of view. The combination of unsupervised neural networks and a context model gives promising results, without creating a bulky overhead on the user-computer interaction. Simple activities like sitting, walking and running are usually recognised within tens of seconds if the accelerometers are placed on the user's leg or hip. For locations, the light sensor has proved to be very efficient, especially when cues such as light-frequency are used. However, as a consequence, this also means that recognition deteriorates as lighting conditions change. Combination of light sensors, GPS and/or beacons would be very interesting in that regard.

In the future, we would like to boost the performance by improving both sensors and cues in both quality and quantity. The experiments up until now used about 10 sensors, but we expect to increase this number significantly. Other important issues we are researching are placement of sensors (on both devices and clothing), the grouping of sensors for the clustering, and redundancy of sensors to make the system truly robust. Finally, the Kohonen map also offers an intuitive representation to the user of how contexts are stored and learned, which is not obvious in machine learning, especially in neural networks.

The framework of this paper was given by the TEA project [3], which is sponsored by the European Commissions "Fourth Framework". Thanks go out to the people from all project-partners: Starlab Research (Belgium), Nokia Mobile Phones (Finland), TecO (Germany) and Omega Generation (Italy).

References

1. Chen D, Schmidt A, Gellersen HW. An architecture for multi-sensor fusion in mobile environments. In: Proceedings of the International Conference on Information Fusion, Volume II. Sunnyvale, CA, USA, July 1999; 861–868
2. Schmidt A, Aidoo KA, Takaluoma A, Tuomela U, Van Laerhoven K, Van de Velde W. Advanced interaction in context, In: Gellersen H (ed) Handheld and Ubiquitous Computing, Lecture Notes in Computer Science. No. 1707. Springer-Verlag, Heidelberg, 1999; 89–101
3. Esprit Project 26900. Technology for enabling awareness (TEA). http://tea.starlab.net
4. Mitchell TM. Machine learning. McGraw-Hill, 1997
5. Kohonen T. Self-organizing maps. Springer-Verlag, Heidelberg, 1997

Correspondence to: Kristof Van Laerhoven, Starlab Research NV/SA, St-Michielslaan 47, 1040 Brussels, Belgium. Email: kristof@starlab.net