Experimental environment to learn user preferences.

Alejandro Fernández-Montes[1], Juan A. Ortega[1], Luis González[2], Juan A. Álvarez[1]
[1]Department of Computer Science, University of Sevilla, Sevilla, Spain
{afernandez,ortega}@lsi.us.es jaalvarez@us.es
[2]Department of Applied Economics, University of Sevilla, Sevilla, Spain
luisgon@us.es

Abstract

The automation of smart environment systems is one of the main goals of smart home researching. This paper focus on learning user lighting preference, considering a working field like a standard office. A review of the smart environment and devices setup is done, showing a real configuration for test purposes. Suitable learning machine techniques are exposed in order to learn these preferences, and suggest the actions the smart environment should execute to satisfy the user preferences. Learning machine techniques proposed are fed with a database, so a proposal for the vectorization of data is described and analyzed.

1 Introduction

Smart home technologies are often included as a part of ubiquitous computing. Mark Weiser [Weiser, 1999] outlined some principles to describe Ubiquitous Computing (ubicomp) from which we emphasize that the purpose of a computer is to help you do something else.

Home technologies have tried to help home inhabitants since its creation. Nowadays, due to the popularization of computational devices, ubiquitous computing is called to be the revolution to develop smart systems with artificial intelligence techniques.

This article focuses on modeling smart spaces to apply machine learning techniques. We have focused in the learning of user preferences for the lighting of a space. In order to interact with the space and retrieve these preferences, an office at the department of Computer Languages of the University of Seville has been provided of several devices to accomplish these tasks.

Artificial intelligent methods can be supported by this model like machine learning algorithms where is centered this article. Some techniques are presented to accomplish this goal in section 4.1. Finally we propose some expansions which could be studied in order to cover other cases.

2 Experimental environments

Two different environments are used for data collection. The first one is a simulated environment developed in Java that allows researchers to generate simulated and synthetic databases.

Second one is a standard office at the department of Computer Languages is used for data collection during the experiments. This office is intended to be used by a single person, who could be eventually visited by other work mates or students. It is illuminated by natural light from the window and four artificial fluorescent lights which can act as a complement of the natural light or like unique source of light.

Figure 1 shows the distribution and setup of the room.

As you can see, the orientation of the window is south, so it maximizes the quantity of light that receives during a day. This fact must be considered when analyzing results.

3 Related devices

In this section we present the devices which will interact in the setup proposed. The concrete model and manufacturer of the devices are detailed although the learning system should be independent of these details.

3.1 Sentilla Tmote

Sentilla Tmotes are the devices which detects the quantity of light. In this setup we propose to install one mote indoor, and another outdoor. This way we can compare the preferences of the user indoor with the quantity of light outdoor. We'll be able to deduct some weather parameters in real time too. The dimensions are 8 cms. of width and 3.2 cm. of height, so it is quite small to suit ubiquitous applications and non intrusive systems. These devices also implement a humidity and temperature sensor, which could be requested for future improvement and expansions. Figure 2 shows the Sentilla *Tmote* module.

The connectivity with other motes and computers is done through the IEEE 802.15.4 (ZigBee) protocol, which minimizes battery consumption. Zigbee supports mesh networking so this is the topology we will adopt and this way the *motes* can create a network to share information and forward it to reach wider areas.

Nowadays Sentilla *Tmotes* are packaged in a beta development kit, including an IDE based on Eclipse 3.2 for developing. The hardware implements a Java Runtime Environment which can run different applications to retrieve, process and send information from sensors.

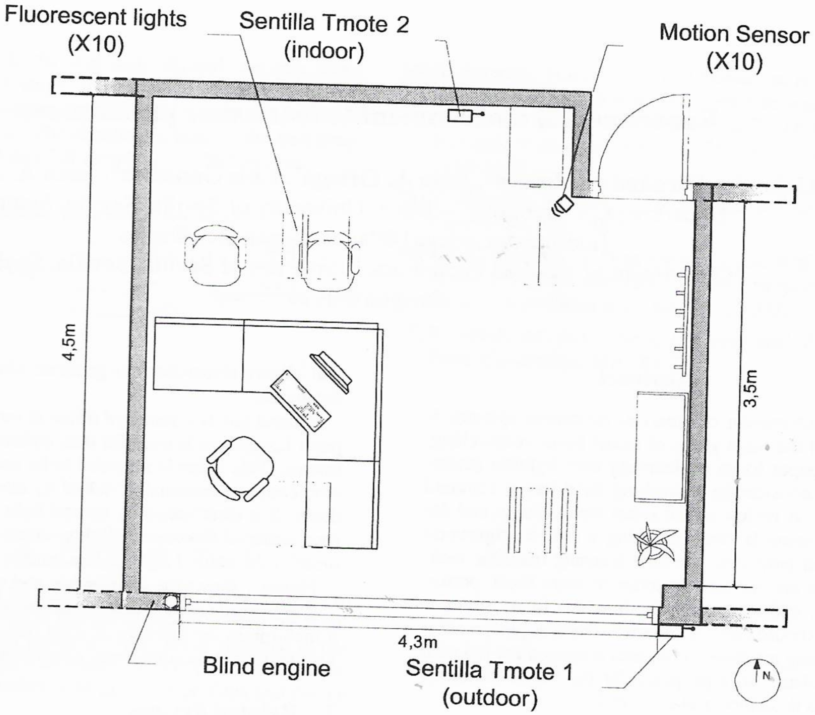

Figure 1: Room setup.

3.2 Motion Sensor

A motion sensor is indispensable to determine when a user is at the room. This could act as a trigger of the learning algorithm to retrieve, process and send the information from sensors.

Domoweb project implemented an OSGi platform with software components to interact with X10 devices. The figure 3 shows the MS13A, a wireless device which interacts with a gateway that routes wireless messages over the electrical cable using X10 protocol. Notice that, although X10 protocol is an old-fashioned technology, it carries out its purpose perfectly.

3.3 Fluorescent light

These lights are activated with an X10 actuator, like the Appliance module AM486 in order to determine when a light has been switched on and off, and its current state.

3.4 Blind engine

Lighting preferences are directly related with the quantity of light that goes through the window. Therefore the state of the blind or curtain will affect the lighting of the room. Some devices are under study but at the moment of writing this article none satisfied our requirements of wirelessly communication and standardized protocols.

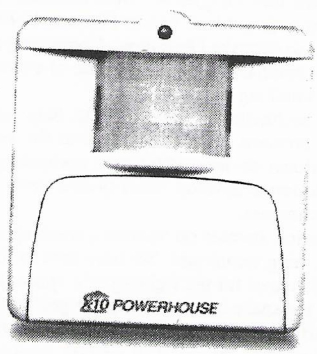

Figure 3: Front of the MS13A. X10 Wireless Motion Sensor.

4 Framework of learning

As exposed before, our goal is the learning of the lighting preferences of a single user. This machine learning is done over the statistical data retrieved at the environment shown in sections 2 and 3. In general, we will retrieve a set of data X called *input*

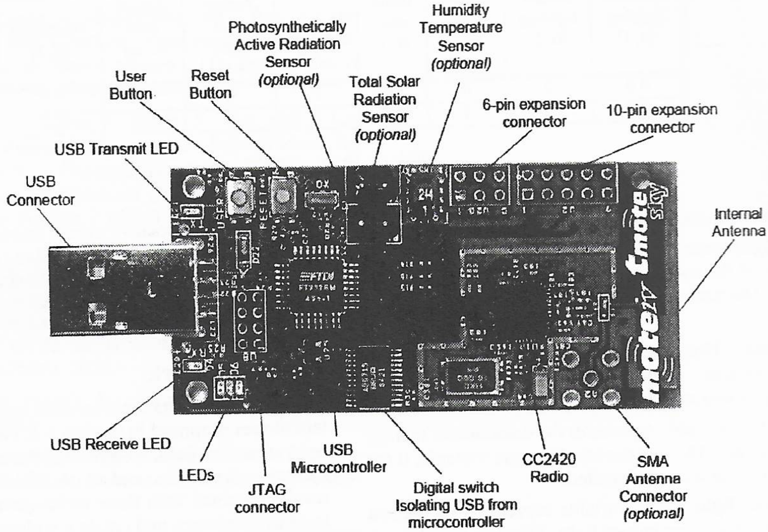

Figure 2: Front of the *Tmote* module.

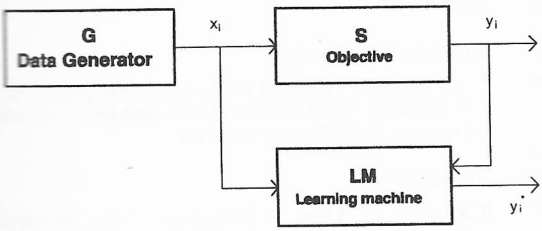

Figure 4: Learning model.

Figure 5: .

$$X = \{x_1, x_2, \dots, x_n\}$$

and we will have a set of data Y called *output*.

$$Y = \{y_1, y_2, \dots, y_m\}$$

The process of learning tries to search a functional dependence between both sets of data. The framework is based on V.N. Vapnik [Vapnik, 1998] model of learning with examples. The model is composed by three elements as shown in the figure 4:

1. **Data generator**. Samples of X, retrieved by the infrastructure proposed.
2. **Objective** (aka supervisor). User preferences.
3. **LM**. Learning machine.

The learning can be carried out due to the dependence of the user preferences with his habits. Normally we have the same lighting preferences. These preferences must be learned at every environment, due to its dependence with the location, orientation of the window, devices setup and so on.

4.1 Techniques

Two families of algorithms, related with learning machine, can be considered although both are going to be supervised.

Support vector machine (SVM) and Neural networks (NN) are the selected techniques to learn user lighting preferences. The main advantage of these techniques is that always offer an output. On the other hand it is hard to interpret or understand their outputs which is an important feature that prediction algorithms should implement as expounded in [Fernández-Montes *et al.*, 2007].

The other family of algorithms considered is the machine learning techniques based on rules. These algorithms provide an output easier to understand and interpret, but their main disadvantage is that if no rule matches current state, these algorithms don't offer an output.

4.2 Input and output

Table 4.2 shows the input variables X proposed and two sample input data:

- **Outdoor lighting**. This variable represents the quantity of light received from the outdoor Sentilla *Tmote* sensor. Continuous variable from 0 to 1.
- **Indoor lighting**. This variable represents the quantity of light received from the indoor Sentilla *Tmote* sensor.

Outdoor lighting [0, 1]	Indoor lighting [0, 1]	Indoor light state {0,1}	Blind state [0, 1]	Motion {0,1}	Action over light {-1,0,1}	Action over blind [−1, 1]	Threshold [0, 1]
0.9	0	0	0	1	0	0	0.5
0.9	0.7	1	0.5	1	1	0.5	0.5

Table 1: Input sample.

Continuous variable from 0 to 1.

- **Indoor light state**. This variable represents the state of the lights of the room received from the X10 appliance module. Discrete variable, 0 for lights off, and 1 for lights on.
- **Blind state**. This variable represents the state of the blind or curtains. Continuous variable from 0 to 1. 0 for totally closed and 1 for totally open.
- **Motion**. This variable represents the detection of motion sent by the MS13A X10 device. Discrete variable, 0 for no motion, 1 for motion detected.
- **Action over light**. This variable represents the action done by the user over indoor light. Discrete variable, -1 lights switched off, 0 no action, +1 lights switched on.
- **Action over blind**. This variable represents the action done by the user over the blind/curtains. Continuous variable, -1 means the user closed it totally, 0 no action, +1 means the user opened it totally.
- **Threshold**. Represents the current user lighting preference. 0 represents minimum room lighting, 1 represents maximum room lighting.

Notice that Sentilla *Tmotes* offer the quantity of light received in luxes. We have to standarize this data to a [0, 1] interval.

Table 4.2 shows the output variables Y proposed and two sample output data:

Action over light {0,1}	Action over blind [−1, 1]	Action over threshold [−1, 1]
0	0	0
0	0	α

Table 2: Output sample.

- **Action over light**. Represents the action over indoor light predicted by the learning machine in order to satisfy user lighting preferences. Discrete variable, -1 lights switched off, 0 no action, +1 lights switched on.
- **Action over blind**. Represents the action over the blind/curtains predicted by the learning machine in order to satisfy user lighting preferences. Continuous variable, -1 means the machine closed it totally, 0 no action, +1 means the machine opened it totally.
- **Action over threshold**. Represents the correction the machine must done in order to adapt current user threshold. Negative values reduce threshold, 0 represents no action, and positive values represent an increase correction over threshold. The value α of the correction must be determined with care, in order to avoid infinity jumps around user preference, and converge to the real user preference.

5 Future work

Comparative results must be done between machine learning techniques proposed in section 4.1. Next step should focus in enlarge action field, to other rooms with different users, locations, orientations, and so on. This way we could compare results obtained with these techniques in different (but similar) environments, and create a wider *motes* mesh network.

Other field of action could be applying these techniques and algorithm to learn user preference over conditioning. Sentilla *Tmote* devices also include sensors to retrieve temperature and humidity, useful to learn conditioning preferences.

Acknowledgments

This research is supported by the MEC I+D project InCare. Ref: TSI2006-13390-C02-02 and the Andalusian Excellence I+D project CUBICO Ref: TIC2141.

References

[Choi *et al.*,] J. Choi, D. Shin, and D. Shin. Research on Design and Implementation of the Artificial Intelligence Agent for Smart Home Based on Support Vector Machine. *Lecture notes in computer science*.

[Cook *et al.*, 2006] D.J. Cook, M. Youngblood, and S.K. Das. A multi-agent approach to controlling a smart environment. *Lecture notes in computer science*, 4008:165, 2006.

[Das and Cook,] S.K. Das and D. Cook. Designing Smart Environments: A Paradigm Based on Learning and Prediction. *Mobile, Wireless, and Sensor Networks*.

[Fernández-Montes *et al.*, 2007] A. Fernández-Montes, JA Álvarez, JA Ortega, MD Cruz, L. González, and F. Velasco. Modeling Smart Homes for Prediction Algorithms. *Lecture Notes in Computer Science*, 4693:26, 2007.

[Hagras *et al.*, 2004] H. Hagras, V. Callaghan, M. Colley, G. Clarke, A. Pounds-Cornish, and H. Duman. Creating an Ambient-Intelligence Environment Using Embedded Agents. 2004.

[Jiang *et al.*, 2004] L. Jiang, D.Y. Liu, and B. Yang. Smart home research. *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*, 2, 2004.

[Leake *et al.*,] D. Leake, A. Maguitman, and T. Reichherzer. Cases, Context, and Comfort: Opportunities for Case-Based Reasoning in Smart Homes. *Designing Smart Homes. LNCS (LNAI)*, 4008:109–131.

[Li *et al.*, 2006] J. Li, Y. Bu, S. Chen, X. Tao, and J. Lu. FollowMe: On Research of Pluggable Infrastructure for Context-Awareness. *Proceedings of the 20th International Conference on Advanced Information Networking and Applications-Volume 1 (AINA'06)-Volume 01*, pages 199–204, 2006.

[Roy *et al.*, 2006] N. Roy, A. Roy, and S.K. Das. Context-Aware Resource Management in Multi-Inhabitant Smart Homes: A Nash H-Learning based Approach. *Proc. of 4th IEEE Int'l Conf. on Pervasive Computing and Communications (PerCom2006)*, 2006.

[Vapnik, 1998] V.N. Vapnik. *Statistical learning theory*. Wiley New York, 1998.

[Weiser, 1999] Mark Weiser. The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3(3):3–11, 1999.

[Yamazaki, 2006] T. Yamazaki. Beyond the Smart Home. *Proceedings of the 2006 International Conference on Hybrid Information Technology-Volume 02*, pages 350–355, 2006.