A mixture of experts for learning lighting control

Stephan K. Nufer$^{(1,2,3)}$, **Mathias Buehlmann**$^{(1,2,3)}$, **Tobi Delbruck**$^{(1)}$ and **Josef M. Joller**$^{(2)}$

Abstract. Conventional building automation is preprogrammed with nominally-optimal behavior. Machine learning offers the possibility of learning behavior that is better matched to individual occupant desires and that further reduces energy consumption. However, traditional machine learning is difficult to apply due to 1) the extremely sparse training input–typically 2-3 effector uses/day and 2) to the extreme necessity to avoid occupant rejection by annoying incorrect behavior. Our building[1] is equipped with a LonWorks building automation network that provides sensor (light, presence, temperature, etc) and effector (light and blind switches) information. Previous work using our automation installation [9] and [12], have explored machine learning of user preferences, with the aim of increasing comfort while reducing energy consumption. All the prior systems were rejected by normal occupants. In the work described here we demonstrate for the first time a system for lighting control that has been accepted by normal building occupants. It has been running continuously for the past 70 days in three normal offices occupied by 9 people. It is based on a new multiagent OSGI-based infrastructure and uses Weighted-Majority mixture of experts, to learn user preferences for lighting starting from a tabula rasa state.

1 INTRODUCTION

A major obstacle for autonomously learning dynamic space behaviors and configurations is that sensors perceive and react very differently from the human brain. For instance, we perceive fog with a different luminance intensity than measured by sensors. Furthermore, ambient light levels change dramatically even with small atmospheric changes such as a momentary scattering of particles over the sun. Hence, different skylight levels can be found even under the same sunlight condition [8]. Consequently, intelligent buildings (IBs) need to be flexible enough to quickly react and to respond to such environmental changes. Additional difficulties are created by sparse and sometimes inconsistent user instructions. These make learning behaviour problematic.

2 RELATED WORK

In [9], [10] and [12], a hierarchical fuzzy system approach has been introduced where the inputs to the learning process are real valued variables acquired from sensors. The output of the proposed learning algorithm is a model consisting of a number of a fuzzy rules which are continuously generalized or amplified into a definitive fuzzy rule set using the inductive fuzzy learning algorithm of Castro *et al*. [1] in order to overcome the curse of dimensionality. These fuzzy rules are then used by a fuzzy logic controller (FLC) to take decisions. Feedback acquired from the environment is continuously used by the learning process to adapt the fuzzy logic rules.

Questionnaires answered by previous users have shown that the system was rejected by the occupants and attempts had been made to circumvent it. We believe that the applied approach is not suitable enough to control non-stationary spaces due to the dependence on predefined membership functions. Correspondingly, every space which is unique in its own kind, i.e. west facing window, would need its own set of membership functions to be defined a priori and therefore does not differ from manually calibrating sensors. Membership functions should change over time and thus makes this approach not sufficiently adaptive.

There is a variety of related projects being conducted at other places. For example [5] describe a system where different sensors of a room are connected via a sensor network (LonWorks) to a single embedded agent that is located physically in a room. This agent uses the information acquired to learn fuzzy logic rules with a genetic algorithm learning paradigm. There learning procedure requires explicit feedback of the user. Our work is different because we use a learning algorithm which doesn't require any explicit feedback by the users at all and is also not based on fuzzy logic.

3 OUR APPROACH

Our measurements have shown that some rooms (labs and regular offices) have simple predictable regularity in them which can be learnt by providing a few significant input variables (i.e. interior daylight or daytime) only. However, not all spaces are of such simple structure and ambient needs and thus need to be learnt and controlled using more sophisticated learning algorithms and also more sensory information. However, it is quite difficult to determine and to evaluate the goodness of a learning approach without having the ability to compare them against others. Additionally, not every approach turned out to succeed all of the time [7]. For instance they might only be accurate under specific sensor conditions, determined by the dynamic source of lighting, temperature, humidity, etc. Furthermore, an environment with frequently changing occupants yields an additional difficulty that must be overcome; the ability to adapt to new occupant preferences that ultimately demands that an algorithm must be capable to discard previously learnt knowledge. Our approach [7] in setting up a suitable learning infrastructure introduces a novel Intelligent Building Framework that composes a set of independent light controller agents (LCs). Each LC deals with multiple input dimensions and controls an individual light on a local basis rather then global (i.e. room or even building).

The core of each LC is a Weighted-Majority based algorithm that incorporates a mixture of experts, which we demonstrate succeeds in learning and controlling different spaces. The major benefit by

[1] Institute of Neuroinformatics, ETH Zurich/University of Zurich, www: "http://abi.ini.unizh.ch", email: snufer@ini.phys.ethz.ch
[2] University of Applied Science, Rapperswil
[3] These authors contributed equally to this work.

applying this approach is that each algorithm within each LC will contribute its individual decision, which is weighted by a function of user interactions and the deviation to the target output. For instance, when occupants frequently change, a different algorithm may score better results then an algorithm that is more reliable in steady spaces where the learning and the control can be improved by remembering occupant behaviors that have happened in past, infrequent situations i.e. in foggy days.
The rest of the paper is organized as follows. Section: 3.1 describes the basic LC algorithm, section: 3.2 motivates the need for short and long term memory and section: 3.3 describes the base learners used in the LC algorithm. The test environment is described in section: 4, and results of preliminary experiments are summarized in section: 5.

3.1 The LC algorithm

Our LC algorithm considers the changing strengths and weaknesses of different base learners. The algorithm makes a decision by taking a weighted vote among a pool of base learners and learns by altering the weight associated with each of them. Accordingly, learners that perform better will be rewarded and others punished. In the following, we present a basic version of the LC algorithm. At each time step t (typically 10 minutes) the overall LC decision LC_t is calculated:

$$LC_t = \frac{\sum_{i=1}^{n} o_{i_t} \; w_{i_t}}{\sum_{i=1}^{n} w_{i_t}} \tag{1}$$

where o_{i_t} denotes the output of a base learner and w_{i_t} the associated weight. LC_t is a value between zero and one which is commonly thresholded i.e. $LC_t \leq 0.4 \hat{=} LIGHT\ OFF$; $LC_t \geq 0.6 \hat{=} LIGHT\ ON$. The resulting error is then computed using:

$$error_{i_t} = \left| O_t - o_{i_{t-1}} \right| \tag{2}$$

where O_t corresponds to the current light status ($LIGHT\ ON \hat{=} 1$; $LIGHT\ OFF \hat{=} 0$). Reinforcement signals observed by the system within a reasonable time period Δt_{user} (typically 20 minutes) are forwarded to a decaying function that extracts a learning rate α_t that is used to adapt the weights of the base learners:

$$\alpha_t = \alpha_{Max} e^{-c\Delta t_{user}} \tag{3}$$

$$w_{i_t} \leftarrow w_{i_{t-1}} \; (1 + \alpha_t - 2\,\alpha_t\, error_{i_t}) \tag{4}$$

Also, to prevent "successful" algorithms from indirectly punishing one another due to small target error values, a suppressing, sigmoidal function is applied. Hence, if the true error is small the function artificially even lowers the true error value produced by the learner. Reciprocally the function artificially increases the errors for learners that have been off base. Correspondingly, the function either amplifies or damps the extent how strong the weight are being changed.

$$error_{i_t} \leftarrow \frac{1}{1 + e^{-c\;(error_{i_t} - 0.5)}} \tag{5}$$

A basic version of the LC algorithm is given in Alg. 1

3.2 STM (Short Term Memory) vs. LTM (Long Term Memory)

A common problem with online learning which has also been addressed in [12] is to decide when to discard old and when to incorporate new data for the learning (STM vs. LTM). Also, *finding a suitable training-set size* for any learner of an arbitrary space is cumbersome but ultimately essential. Collecting a bunch of data isn't very practical either due to often dealing with an *imbalanced class problem* and above all, building occupants may frequently change.

```
repeat
    Fetch input vector x⃗_t (Sensor values from humidity,
    temperature, daylight, blinds, etc.)
    foreach base learner do
        o_{i_t} = getDecision(x⃗_t)
    end

    // Set the status of the light
    LC_t = (Σ_{i=1}^n o_{i_t} w_{i_t}) / (Σ_{i=1}^n w_{i_t})
    t ← t + 1

    foreach base learner do
        error_{i_t} ← |O_t − o_{i_{t−1}}|
        error_{i_t} ← 1 / (1 + e^{−c (error_{i_t} − 0.5)})
        α_t ← α_{Max} e^{−cΔt_{user}}
        w_{i_t} ← w_{i_{t−1}} (1 + α_t − 2 α_t error_{i_t})
    end
    // Normalize weights
    foreach base learner do
        w_{i_t} ← w_{i_t} / (arg max_i w_{i_t})
    end
    // Clamp weights
    foreach base learner do
        if w_{i_t} < w_{min} then
            w_{i_t} ← w_{min}
        end
    end
until Stop condition reached ;
```

Algorithm 1: LC Algorithm

3.3 Base learners

Applying a mixture of experts for controlling arbitrary spaces therefore seems to be a reasonable solution for our task, since it dynamically tries to discover and adapt to constantly changing needs by consulting a voted expert decision. In the following we will focus on some of the applied base learners. We first examined a variety of learning algorithms and tested them using our own developed real-time simulation software. This software simulates custom spaces as well as fictional occupants with varying preferences. Generally we were interested in the strengths and weaknesses of the algorithms as well as under which conditions they succeed in learning space and changing occupant behaviors.
The core approach that we propose here, addresses some of the issues stated in the previous section: 1) *discover a suitable training-set size* and 2) *the class imbalance problem*. The basis underlies a foregoing data pre-processing step, clustering, that has been used by two (out of five) base learners.
The aim with clustering is to improve any conventional learning algorithm such as ANNs or other regression techniques by preceding additional ambient noise filtering. In other words we use clustering to preserve collected data into categories and thus provide a solution against losing infrequent knowledge.
However, most clustering algorithms such as the *k-means* require a

priori knowledge on the number of clusters and therefore a suitable number of clusters must dynamically be discovered. One of the two base learners applies the g-means algorithm [4], that discovers an appropriate number of clusters using a statistical test for deciding whether or not to split a k-means center into two new cluster centers. Thereby the split is conducted on the centroids whose "member data" appear not to come from a Gaussian distribution. The other base learner also employs a cluster algorithm, GNG [2], [3] that dynamically discovers a suitable number of clusters by producing a set of growing gases (neurons). Hereby the neurons try to best fit our non-stationary model.
Having clustered the data, each of the data points provided by each of the gained clusters can be used by a supervised learning technique such as regular ANNs. The regression is conducted on a local or global basis.

- Local refers to the procedure where the new data is first classified to the most probable cluster. Only the data of this cluster is then used as the sample training-set.
- Global does't classify the new data to a cluster but instead takes all of the clusters as the training-set (See next paragraph).

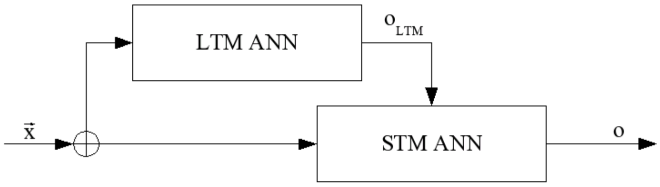
Both the problems: the class imbalance problem and the problem of discovering a suitable training-set size is further addressed by

- Limitting the size of clusters (ring-buffering) to prevent the training-set from constantly growing and also to avoid the training-set to become largely imbalanced.
- The training-set should be re-clustered before any conventional supervised learning technique is employed. Hereby, each the re-clustering step should be launched with an initial set of clusters (Depending on the number of clusters of the previous step). Thus, we allow candidate clusters to be merged with other similar clusters to prevent a form of over-clustering. Also, clusters which only contain a few data members should be deleted completely (outliers).

Our simulations have shown that a global regression is better matched in our building intelligence system since it reduces over-fitting. In summary, we perform clustering to enhance the decision making on distinct spots where conventional algorithms would fail to provide a fast reliable but necessary prediction since most algorithms are quite often not agile enough to react to such changes within a reasonable time.

The other three applied base learners shown in [7] have been realized with conventional techniques. The basic idea of the approaches are given below:

- A simple statistical approach that has shown to succeed in so called *Simple Environments*. Such an environment signifies that learning can be performed by only using the interior daylight and the daytime as the major measure.
- Conventional multi-layered ANNs.
- A form of dual layer networks that simply involves two multi-layered ANNs, each of which working with different training-set sizes. One for representing a short term memory which is affected by the output network of the long term memory (Fig: 3.3). This approach was inspired by the fact that LTM in human brains have a strong influence on the perception through top-down processing. In [7], other similar network variants have also been discussed.



4 TEST ENVIRONMENT

Our novel adaptive building intelligence (ABI) system presented in [7] and [6] is required to run 24/7 and thus some measures have been taken to improve the reliability by applying self-healing upon any software component or networking failure. Further, providing reliable presence in a space that is only equipped with regular PIR sensors is according to [7] very hard to achieve due to the required line of sight with the subjects and thus make energy savings and learning occupant behaviour cumbersome. Hence, PC Presence detectors [11] have been incorporated which helped us to improve the presence detection within our test environments. The building structure here at the INI[1] is a non-stationary environment where not only individual desires are constantly changing but also the physical structure (e.g. seasonal weather, mobile walls in multi-office environments).

5 PRELIMINARY RESULTS

The following real experiment was used to evaluate the success of the learning. Three rooms, each of which comprises 2 LCs, were tested for a period of 70 days. Two rooms are illustrated in Fig. 1. The gap between week four and five in room 55.G.84 is due to Xmas and New Year's eve, where nobody was working. The collected data in-

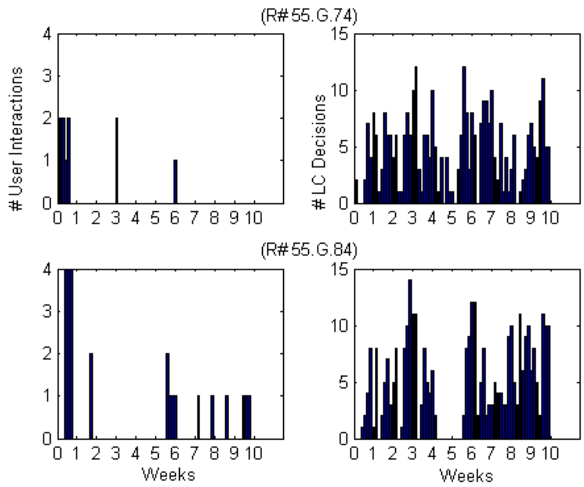


Figure 1. The left plots show the received occupant-, and the right plots the conducted LC decisions, versus weeks.

dicate that user interactions which are normally taken by occupants had been taken over by the system. Thus, the system was assisting its inhabitants in their ordinary daily tasks. In the first week four user interactions were received which decreased to only 1-2 or mostly even zero. The learning was conducted every 10 minutes under the condition that someone was present. Additionally, in order not to end up in opposing interactions a delay of one hour was being enforced upon receiving any system or user interaction. Fig. 2 illustrates how the contribution of the base learners have varied during that 70 period.

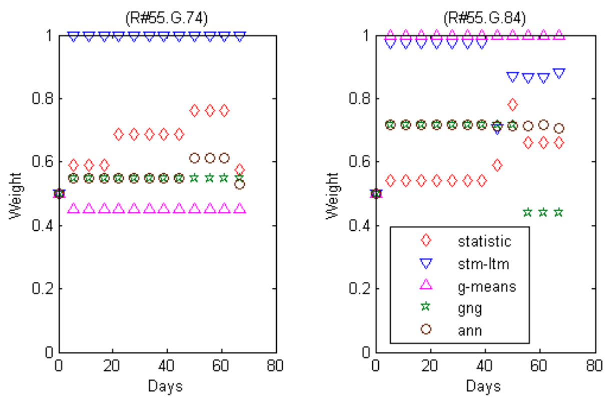


Figure 2. The plots illustrate the varying contribution (weight [0-1]) of the base learners.

Each associated weight reflects how strong its opinion was considered by the overall decision. When comparing Fig. 2 with Fig. 1 we can observe that around day 42 most weight alterations have been conducted due to newly received user instructions. Five learners were incorporated and their weights recorded; Fig. 2, shows that not all base learners performed equally well in our two test rooms, but at the same time, we can observe that the Weighted-Majority mixture of experts succeeded to control the two rooms even though we had been running the same base learners in both rooms.

6 SUMMARY

We have proposed and studied a new approach that succeeded to control the lights in three different rooms for a period of 70 days. Each light controller hereby dynamically considered a mixture of expert decisions. The evaluation of the collected data has shown that the collective decisions updated to individual space functions without a priori knowledge. Also, a stable OSGi-based infrastructure was developed that incorporates the presented IB framework that provides a common generic architecture that facilitates further development. In the future, such a generic setup will be beneficial to be even adapted to other controllers such as blinds. Also, as illustrated, we believe that foregoing clustering can help to improve future decisions where conventional approaches are not agile enough to react to sudden environmental changes within a reasonable time. Although these preliminary results are encouraging, they need to be studied over a longer period of time with a greater variety of rooms, occupants and weather conditions, and controlled rooms need to be quantitatively compared with uncontrolled rooms. This work is ongoing.

ACKNOWLEDGEMENTS

The authors like to thank Rodney Douglas and Kevin Martin for letting us utilize this wonderful infrastructure. We would also like to thank Simone Schuhmacher, Kathrin Aguilar Ruiz-Hofacker, Steven Fry and Giacomo Indiveri for letting us control their room. Finally we would like to thank the referees for their comments which helped improve this paper.

REFERENCES

[1] J. L. Castro, J. J. Castro-Schez, and J. M. Zurita, ‘Learning maximal structure rules in fuzzy logic for knowledge acquisition in expert systems’, *Fuzzy Sets and Systems*, **101**, 331–342, (1999/2/1).

[2] Bernd Fritzke, ‘Supervised learning with growing cell structures’, in *Advances in Neural Information Processing Systems 6*, eds., J. Cowan, G. Tesauro, and J. Alspector, 255–262, Morgan Kaufmann Publishers, San Mateo, CA, (1994).

[3] Bernd Fritzke, ‘A growing neural gas network learns topologies’, in *Advances in Neural Information Processing Systems 7*, eds., G. Tesauro, D. S. Touretzky, and T. K. Leen, 625–632, MIT Press, Cambridge MA, (1995).

[4] Greg Hamerly and Charles Elkan, ‘Learning the k in k-means’, in *Advances in Neural Information Processing Systems 16*, eds., Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, MIT Press, Cambridge, MA, (2004).

[5] Martin Colley Hani Hagras, Victor Callaghan and Graham Clarke., ‘A hierarchical fuzzy-genetic multi-agent architecture for intelligent buildings online learning, adaptation and control’, *Information Sciences - Informatics and Computer Science*, **150**, 33–57, (2003).

[6] Stephan K. Nufer and Mathias Buehlmann, ‘Intelligent, learning system. - a new ABI system built on the open services gateway initiative’, Technical report, University of Applied Sciences Rapperswil, Switzerland and Institute of Neuroinformatics, Swiss Federal Institute of Technology, Zurich, Switzerland. www: http://abi.ini.unizh.ch, (2005).

[7] Stephan K. Nufer and Mathias Buehlmann, ‘Intelligent, learning systems ABI mark II. - a novel approach for learning dynamic space behaviors in a non-stationary environment’, Technical report, University of Applied Sciences Rapperswil, Switzerland and Institute of Neuroinformatics, Swiss Federal Institute of Technology, Zurich, Switzerland, (2005).

[8] N. et al. Ruck, *Daylight in Buildings - A source book on daylighting systems and components*, Lawrence Berkeley National Laboratory, Berkeley (USA), 2000 [LESO-PB-BOOK-2000-004], 2000.

[9] Ueli Rutishauser, J. Joller, and R. Douglas, ‘Control and learning of ambience by an intelligent building.’, *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, **35**(1), 121–132, (2005).

[10] Ueli Rutishauser and Alain Schaefer, ‘Adaptive home automation – a multi-agent approach’, Technical report, University of Applied Sciences Rapperswil, Switzerland and Institute of Neuroinformatics, Swiss Federal Institute of Technology, Zurich, Switzerland, (2002).

[11] A. Fenkart T. Delbruck and R. Zwiker. Using personal computers for enchanced presence detection in building automation, HCI, 2005.

[12] Jonas Trindler and Raphael Zwiker, ‘Adaptive building intelligence – parallel fuzzy controlling and learning architecture based on a temporary and long-term memory’, Technical report, University of Applied Sciences Rapperswil, Switzerland and Institute of Neuroinformatics, Swiss Federal Institute of Technology, Zurich, Switzerland, (2003).