Towards the Intelligent Home: Using Reinforcement-Learning for Optimal Heating Control

Alexander Zenger[1], Jochen Schmidt[1], and Michael Krödel[2]

[1] Fakultät für Informatik, Hochschule Rosenheim, Germany
[2] Fakultät für Angewandte Natur- und Geisteswissenschaften
Hochschule Rosenheim, Germany

Abstract. We propose a reinforcement learning approach to heating control in home automation, that can acquire a set of rules enabling an agent to heat a room to the desired temperature at a defined time while conserving as much energy as possible. Experimental results are presented that show the feasibility of our method.

1 Introduction

By far the most energy in homes is used on heating, one of the main reasons being badly insulated buildings that were constructed before there was any legal obligation to conserve energy. Using home automation technology the energy consumption of buildings can be controlled much more efficiently than by adjusting heating parameters manually. Up to now, mainly model-based methods are used for this purpose, e. g. [2] where a model of environment and building has to be parametrized. Correct modeling is complex and requires expertise on how buildings and rooms behave when heated. As the behavior can change over time due to modifications (e. g., replacing carpet by solid hardwood flooring), the model has to be adjusted accordingly each time. Therefore, adaptable machine learning techniques that do not require model building are an interesting way of dealing with these issues in home automation. Machine learning approaches have been applied in home automation to some degree. For example [4] uses a neural network to learn when inhabitants are at home in order to control residential comfort systems according to their needs. While reinforcement learning is relatively popular in control engineering for designing low-level control units (cf. [1]), to our knowledge, this is the first time reinforcement learning is used in the context described in this paper. Not only is it important that the desired temperature is reached, but also that this is achieved with low energy consumption. The goal is to heat a room, which has cooled-off to a certain temperature, to a defined temperature within a given period of time. This is a typical use-case for office buildings, where the heating is normally ramped down over night to conserve energy. In the morning, when employees arrive at their workplace, room temperature should have reached a pleasant level again. Obviously, as the room cool-off over night, the heating has to be turned on at a certain point: Done too early, this leads to waste of energy and increased heating costs. The optimal moment depends on several factors, e. g., inside and outside temperatures, construction materials, etc., and is different for each room. We propose a system for automatic heating control that acquires a set of rules automatically during normal use. A heating controller is capable of learning actively by executing defined actions. No data collection is required beforehand.

2 Environment Model and Learning

In reinforcement learning, the agent has no a priori knowledge of the environment's behavior. The system learns by trial and error how actions and states are linked. The algorithm used in this paper is SARSA(λ) [5, 6], which learns state-action pairs. We reward the learning algorithm only when the final state is reached, therefore a modified version of this method is used as described in [3], where a history of actions and states is stored in each iteration. At the end of an iteration the history is traversed reversely and the utility values are propagated.

After several iterations the system has learned a set of rules that would always be used from there on, as it is the best rated one. A balance is required between exploitation of already learned actions and exploration of new ones. We apply the ϵ-greedy strategy for exploration, where the present solution is used with a probability of $1 - \epsilon$, and a randomly selected action otherwise. At the start, ϵ should be high, so that new rules can be learned quickly. It can be reduced after some solutions have been found based on the residual error.

The system's performance depends highly on the rewards r_i that the agent receives for its actions. We suggest to use the following reward function:

$$r_i = \begin{cases} (r_{i-1}/\alpha) + 0.01\, r_{i-1} & \text{if target temperature reached and end of duration} \\ -0.01 & \text{if target temperature not reached and end of duration} \\ 0.0 & \text{otherwise} \end{cases} \tag{1}$$

where α is the learning rate. The "duration" is a defined maximum number of actions that the agent can perform. The design of the reward function (1) is based on the observations that (a) less heating cycles must result in higher rewards and (b) when a better solution is found by the learning algorithm just once, it should result in a positive reward that is high enough for leaving the current range of rewards. To compute a reward for a given index, a suitable initial value r_0 has to be defined. It has to be sufficiently small, so that no overflows are generated, but high enough so that it can be represented by the selected floating point data type. We use a double precision type and $r_0 = 0.01$. This allows for up to 316 iterations, while a negative reward can still be balanced even in the worst case.

For evaluation we used two main setups: One setup is a pure software solution, i. e., there is no actual heating mechanism controlled. This allows for fast simulation of heating and cooling processes, which might take hours or even days in a real environment. The second setup (the *Model Room*) consists of a physical small-scale room model as well as an actual heating element and sensors connected to a home automation controller. Only the latter will be described here. As heating a real room to conduct experiments would not lead to reproducible results due to uncontrollable environment conditions, we use a small scale model room for this purpose instead. Its behavior corresponds to the situation in a real building and requires a pre-heating (or pre-cooling-off) period before experiments are conducted to reach the desired initial room temperature. The model room consist of a styrofoam box, which has insulation characteristics similar to those of a real room. The outside environment is simulated by placing the box inside a refrigerator which lets the model room cool-off over time while not heated. Temperature sensors are mounted inside the box and the refrigerator, respectively. A 15 watt light bulb located in the center of the box is used as the heating element. It can be turned off and on by a controller that also collects temperature information and is connected to a Unix server running the AI software. We use a WAGO controller [7], which is a standard component for process or building automation.

3 Experimental Results

To evaluate the presented system, we conducted different experiments; only one will be described here. Each experiment consists of a number of iterations. A single iteration corresponds to a complete run heating up the room starting at a defined initial temperature of 8.0°C to a target temperature of 15.0°C. In each iteration, the agent has a defined number of steps where actions can be taken. The number of steps is called duration further on. The set of possible actions consists of turning the heating element on or off. Every experiment begins with an empty data set, which means the agent has no previous knowledge and thus has to learn everything anew. Rewards are computed according to (1) using r_0 for initialization. The minimum heating cycles in the first iteration are initialized using the duration value of 10 steps. To accelerate and improve the learning process, the ϵ parameter is varied as follows: For the first 100 iterations we increased ϵ to a value of 0.8, thus allowing the agent to explore more. Then ϵ was decreased to 0.01, and the algorithm ran another 123 iterations, thereby letting the agent exploit the recently learned data and generate a stable outcome (100 iterations for learning, and additional 23 iterations to check the validity of the result). Figure 1 (left) shows that the target temperature was reached in about 50% of the cases. The process is depicted in Fig. 1 (right). It shows that the defined temperature is immediately reached and kept after decreasing ϵ at iteration 101. This is because a first solution was already found at iteration 24, and improved at iteration 44 during the intensive exploration phase. As a result the target temperature was reached faster and more often. Further improvement is possible by decreasing ϵ right after the first time a solution was found.

The main advantage of our approach over more traditional model building ones is that it does not require tedious manual adjustments of model parameters, which have to be performed anew for each room, as every one behaves differently.

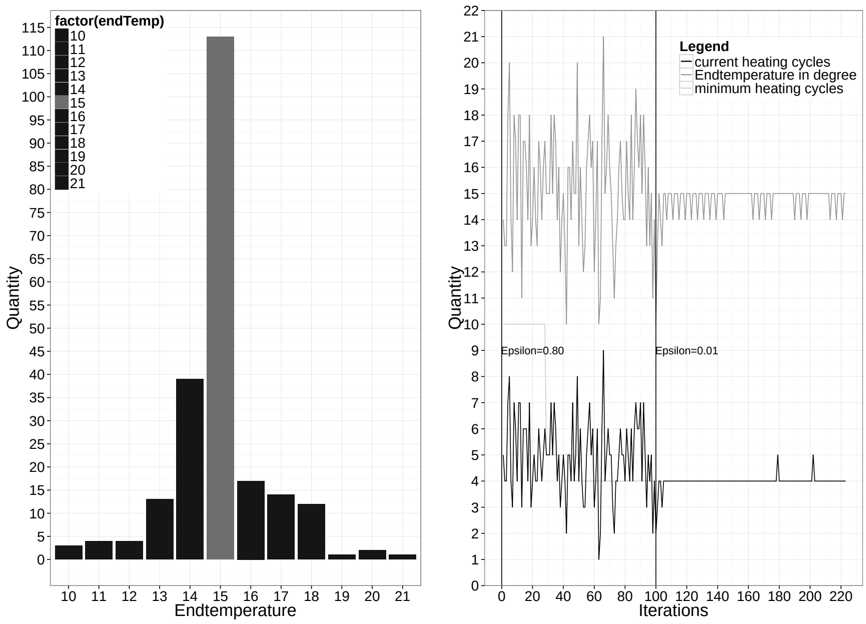

Fig. 1. Left: Reached temperatures histogram. Right: Experiment process

We have presented experimental results demonstrating that the target temperature is reached and kept over time. Future work will include dynamic adaptions of exploitation and exploration phases, as well as further experiments in real environments, in particular normal size rooms.

References

1. Anderson, C., Hittle, D., Ketchmar, R., Young, P.: Robust reinforcement learning for heating, ventilation, and air conditioning control of buildings. In: Si, J., Barto, A., Powell, W., Wunsch, D. (eds.) Learning and Approximate Dynamic Programming, chap. 20, pp. 517–534. D., John Wiley & Sons (2004)
2. Ellis, C., Hazas, M., Scott, J.: Matchstick: A room-to-room thermal model for predicting indoor temperature from wireless sensor data. In: Proc. of IPSN 2013 (2013)
3. Krödel, M.: Autonome Optimierung des Verhaltens von Fahrzeugsteuerungen auf der Basis von Verstärkungslernen. Ph.D. thesis, Universität Siegen, Germany (2006)
4. Mozer, M.C.: The neural network house: An environment that adapts to its inhabitants. In: Coen, M. (ed.) Proc. of the AAAI Spring Symposium on Intelligent Environments. pp. 110–114. AAAI Press (1998)
5. Rummery, G.A., Niranjan, M.: On-line q-learning using connectionist systems. Tech. rep., University of Cambridge, Department of Engineering (1994)
6. Sutton, R.S., Barto, A.G. (eds.): Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning). The Mit Press (1998)
7. WAGO Kontakttechnik GmbH & Co. KG. http://www.wago.us