Hierarchical Preference Learning for Light Control from User Feedback

Amir Khalili, Chen Wu and Hamid Aghajan
Ambient Intelligence Research Lab
Stanford University, Stanford, CA
akhalili@stanford.edu

Abstract

We propose a system for optimized light control in smart homes considering both energy efficiency and user preference. The method is based on learning the user preferences online and under different states (time, location, activity). To achieve adaptive and interactive learning of user preferences, we propose to use hierarchical reinforcement learning (HRL) to adapt the user model dynamically from user feedback. The input to HRL is user’s activity obtained from a two-level vision analysis from a camera network. It includes the user’s position, coarse-level activities (standing, sitting, lying) as well as fine-level activities (reading, eating, cutting, etc.). HRL learns user’s preferences when the user gives feedback to the system through changing the offered light setting. The strength of HRL compared to regular reinforcement learning is that it is able to first learn preferences for a predefined “groups” of states (such groups are called routines in our paper). The number of routines is significantly smaller than the number of states, therefore the convergence can be significantly expedited. As more feedback is given by the user, HRL refines the preferences for individual states within the routines. The optimal light intensity level is determined as a balance between user satisfaction and energy cost.

1. Introduction

There has been extensive research on smart homes with a variety of applications including monitoring systems for independent living of elderly and accident detection [7], smart appliances/devices such as refrigerators, and automated services such as energy and light control [12,13]. The goal is to provide services that maximize the user’s comfort while minimizing the user’s explicit interaction with the environment as well as the cost of the service. In some services such as light and air conditioning control, the user comfort and the cost of energy are jointly considered to find a balanced operating point. This type of tradeoff is relevant as considerations for energy efficiency are finding a major influence on the design of smart buildings. Another type of service may be designed mainly to provide a comfortable ambience for the user. For example, in the Philips Experience Lab [3], LED lights of different colors are installed at several locations in the room, and different themes of ambient lighting can be applied depending on the user’s preference. In most current applications, the system provides its services either based on a set of embedded fixed rules which determine the service parameters based on the situation, or based on a query from the user every time a change is to be applied to the service. For example, a simple light control system may be programmed to turn the lights off automatically when the user sits to watch TV. Without an intelligent method to adapt the settings to the user’s preferences, this may result in a rigid operation which may not be desirable for many users. Or the system may ask the user to confirm the action every time the same event is detected. When different services are to be offered throughout the day such queries disrupt the user’s normal routines and may render the system obtrusive.

In this work, our goal is to create a user-centric methodology for adaptation of system services based on both real-time context, namely the location and activity of the user, and accumulated knowledge about the user preferences under different activity contexts. These preferences are learned through user’s explicit or implicit feedback to the system when the user opts to react to the provided service. As a result, the system adapts to provide the most satisfactory level of lighting to the user according to the location and activity type.

Learning user’s preference of the service parameters requires information of the user’s state. In many smart home system embodiments, readings from different sensors are employed to infer user’s activities [8]. Motion sensors, light sensors and sensors that indicate user’s interactions with objects have been used for this purpose. In a variety of cases such sensory data may not provide sufficient detail for inferring the activity. For example, the user may watch TV, read newspapers or talk with the family in the living room, and the type of lighting or background music he may prefer un-
under each situation may likely differ. In our system a network of cameras is used with the objective of obtaining a rich description of the user’s location, pose and activity. This enables the system to learn and correlate the user’s preferences with more specific contextual information about the user.

The rest of the paper is organized as follows. In Sec. 2 the architecture of the proposed system is presented and the methods implemented in the context module to acquire the user’s state and in the adaptation module to obtain and adapt to the user feedback are described. Sec. 3 presents the result of experiments on learning and adapting to the user preference model. These results include an analysis of a layered vision-based user activity classification module as well as a study of the adaptation behavior of different learning methods applied to the light control application. Finally, Sec. 4 offers some concluding remarks.

2. Context-based Light Control with Hierarchical Reinforcement Learning

The goal of our work is to provide a comfortable light setting to the user according to the observed activity and the user’s stated preferences while minimizing energy cost. The system consists of three main modules: the context module which is in charge of inferring user’s state and maintaining a model for user profile, the adaptation module which adapts the system’s functional parameters through user feedback and learning, and the actor module which balances the tradeoff between user comfort and energy usage (Fig. 1). In the context module, the user’s location and activity are analyzed from visual data from a camera network. A two-level activity analysis is used to obtain fine-grained classification of activity. In the adaptation module, hierarchical reinforcement learning is applied to learn the user’s preferred light settings. These settings are represented by a utility function estimated from the adapted reinforcement learning Q-tables, which carry preferred values for a range of user states across location, activity, and time. This learning process is online and adaptive. With the learnt satisfaction utility functions, the actor module solves for the optimal light setting through an optimization method that aims to maximize the user’s satisfaction while minimizing energy consumption.

2.1. Hierarchical activity analysis with a camera network

Different types of sensors have been used to sense user’s activities in a smart environment. Examples include state-change sensors [15] attached to appliances and RFID tags and readers used with household items [14] to collect object usage data as an indirect way to infer user activity. Logan et al. in [9] study activity recognition with a variety of sensors including RFID sensors, switch sensors, and motion sensors, and offer an evaluation in real-world conditions. They show that with the named sensors it is difficult to detect fine-grained activities such as “reading” and “eating”. They also state that visual sensing provides more information which is oftentimes complementary to other sensors. Vision-based human activity analysis has seen significant progress in recent years [10], and examples of classifying fine-grained human activity based on video can be found, for example, in [13].

We use a hierarchical approach to classify user activities with visual analysis in a two-level process. Different types of activities are often represented by different image features, hence attempting to classify all activities with a single method would be ineffective. In Fig. 2 activities are represented by coarse and fine levels. The coarse activity level includes the classes of standing, sitting and lying, which relate to the pose of the user. The fine activity level consists of activities involving movement such as cutting, eating, reading, etc. We apply such a hierarchical approach because the first-level activities are discriminated based on pose, while the second-level activities are classified based

Figure 1. Core modules of the system.

Figure 2. Hierarchical activity analysis.
on motion features.

In the first level, activity is coarsely classified into standing, sitting and lying with temporal conditional random fields (CRF), through employing a set of features consisting of the height of the user and the aspect ratio of the user’s bounding box. The foreground bounding box of the user is extracted with adaptive background subtraction. Position and height of the user are calculated from the bounding boxes using calibration parameters of the camera network. Based on the result of the coarse level, the activity is further classified at the fine level based on spatio-temporal features \([13]\). K-means with \(N\) clusters is used to cluster the features, and each feature is quantized into the index of the closest cluster. Bag-of-features are collected for every \(t\) second duration. Finally SVM is used as the classifier. The semantic location context constrains activity types for classification in the fine level (Table 1).

### Table 1. Semantic location context for activity classes.

<table>
<thead>
<tr>
<th>location</th>
<th>activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>kitchen</td>
<td>cutting, scrambling, vacuuming, others</td>
</tr>
<tr>
<td>dining table</td>
<td>eating, vacuuming, others</td>
</tr>
<tr>
<td>living room</td>
<td>watching, vacuuming, others</td>
</tr>
<tr>
<td>study room</td>
<td>computer, vacuuming, others</td>
</tr>
</tbody>
</table>

#### 2.2. Reinforcement learning

A challenge faced by adaptive methods to learn the user’s preferences is that the user may change his behavior over time \([3]\). Therefore it may not be practical to learn the user’s preference off-line and instead a continuous learning algorithm is needed. A case study on learning and predicting entertainment preferences of children playing physical games is presented in \([19]\), where four preference learning algorithms are compared. In \([3]\) the authors present the state-of-the-art in machine learning techniques for learning user patterns in intelligent environments, and propose ideas on combining different methods based on the complexity of the problem and the strengths of different techniques.

We apply the framework of reinforcement learning (RL) to adapt to the user preferences under different contextual conditions. RL as an unsupervised algorithm does not need predefined models of the service parameters and learns the model from the user’s feedback to the service the system provides. The state of the user (time of the day, location, and activity) is observed by the system, and the service is chosen automatically by the algorithm based on the user’s preference learned up to that time. If the user is not satisfied and applies a change to the service parameters using the system’s user interface or simply leaves the area, the event is recorded as explicit or implicit feedback respectively, and is used to update the decision policy.

Reinforcement learning was used by Mozer and Miller \([11]\) to control a home’s lighting system according to the user movements and based on user preferences. In their work, the penalty was defined according to the actual energy cost and degree of dissatisfaction of the user. Given the penalty, the decision learner explores different light patterns and estimates their utility under the prevalent user movement behavior. While Mozer’s system does not assume a prior definition of the environment, Zanderberg et al. \([20]\) set a starting point for the system according to prior assumptions by the designer. Starting from a pre-defined setup, the system begins to explore the best control policy to maximize the user’s satisfaction. In \([4]\), Crites and Barto designed an intelligent elevator with nearly optimized movement strategy for an office building. They demonstrated that a reasonable performance could be achieved even without a complex algorithm. In \([5]\) the authors simulated a smart home environment and used a reinforcement learning method to discover user’s daily behavioral pattern in a hierarchical structure to generate automatic control policies.

Our work differs from the mentioned methods in the following aspects. First, hierarchical reinforcement learning (HRL) is used to overcome the slow convergence issue of regular RL methods. In our formulation, the user’s preferences are first learned for routines consisting of a group of states. If more observations are available through time, preferences are learned gradually for each state within the routines. The faster convergence rate offered this way by HRL makes the system practical for real applications. Second, a utility function relating user’s satisfaction with light intensity is adapted from its initial design-stage setting as the user provides feedback. The utility function is employed to estimate the user’s preference under the similar but not yet observed situations. Third, while adapting to the user’s preferences the system also minimizes the consumed energy. The optimal light setting is determined through an optimization formulation that balances the user’s satisfaction and energy consumption.

#### 2.2.1 HRL model for adaptive light control

As the inference model for user’s activities is defined in a layered fashion based on subsequent classifications in coarse and fine levels, the learning of the user’s preferences can also be modeled in a hierarchical fashion. This enables the more frequent updates based on either the coarse activity type or location context to be used for adjusting the utility functions for a broader range of similar user activities. For example, similar light level utility functions can be adjusted for different user activities in the kitchen when the system receives a feedback from the user when he is in the kitchen. Or when the user provides a feedback on the light level set when he is observed reading, the feedback is used to update all utility functions that relate to the reading.
activity even in other location contexts. This abstraction in hierarchical levels shrinks the space domain and can significantly expedite the search for optimal or suboptimal decision policies for states related to the updating group. In the absence of the state abstraction, the proper decision would need to be explored, examined and learned for each of the user states, causing a long convergence time for complex problems. Value function approximation methods (such as tile coding [16]) in which the value of a state-decision pair is approximated by neighbor states are employed to overcome the convergence problem. State space abstraction in RL using the hierarchical nature of the problem was explored in MAXQ Hierarchical Reinforcement Learning (HRL) [6]. MAXQ-HRL uses prior knowledge of hierarchical behaviors whose optimal policies can be learned simultaneously.

In our model, the optimal regularization of the lighting system depends on the activity and location of the user. The adaptation module keeps a list of valid activities in each context. For example, “computer”, “reading” and “vacuuming” are only valid in the “study room” as illustrated in Fig. 3. The location context is used as a hierarchical abstraction level for adaptation. Transitions between activities within a location is modeled through a finite Markov process. Valid activities for different locations are shown within the boundaries of Fig. 3 and the connecting lines correspond to the finite Markov process model. We call the finite Markov process of each location as a “routine”. The length of observations of a routine can vary. In Fig. 4, the black nodes in the lower level are the activities, and white nodes in the higher level are routines.

The routines $M_i$ in high level also can be described as a Markov process as shown in (Fig. 5). The transition function between routines depends on the probability distribution $\mu(location\{s_t\}, M_i)$ and transition probability of $\delta$. The values of transition probability $\delta$ for adjacent places is learnt from observation of the user. For non-adjacent places $\delta = 0$. Probability distribution of $\mu$ shows the probability of an activity in a location. As an instance, the probability of “reading” at study room is more than its probability at living room. The system selects a routine $M_i \in \mathcal{M}$, according to the high-level policy based on $\mu(location\{s_t\}, M_i)$ and $\delta$. Each $M_i$ has a termination condition $T_i$. When $T_i$ is met, a new routine $M_j$ is activated. In general the termination condition depends on the sequence of observed state-action pairs. In our method, termination of a routine is conditioned on the location of the user.

Each routine $M_i$ has its own policy $\pi_i$ to control the lighting system using actions based on user’s preference. Such policies are called “flat” or “low-level” policies. The policy is trained using MAXQ which is a hierarchical adaptation method of reinforcement learning. To this purpose we assign a finite-state machine to each routine. The optimized actions are selected according to the current state of the system $s_t$ and the activated routine $M_i$. The actors only receive primitive actions derived by the low-level policies.

2.3. Utility function as user profile

The system learns the user’s light setting preference under different situations in order to create a user profile. The Q-table is a tabular profile in reinforcement learning which records inclination of selecting actions at different states. In Fig. 6 each row $s_t$ corresponds to a state, and each column $l_j$ denotes a light setting. The value in the tabular refers to the score of level $l_i$ at state $s_j$ as expressed by the user.
Each user in the smart home may have his own Q-table. Uploading a table customizes the system services for that particular user. The Q-learning operation adapts the profile to user preferences smoothly by changing the weights and probability of selections.

Although the user’s preferences are learned online and encapsulated in the Q-table during the reinforcement learning process, the Q-table may remain sparse before the system has collected enough information of the user about his preferences for all the possible states. In order to enable the system to estimate the user’s preference for any light setting \( l_i \) at any state \( s_j \), while user scores are still not learned, we estimate a utility function \( U(s, l) \) given the current form of the Q-table. The idea is based on the assumption of continuity in the scores for nearby light settings, so the scores can be estimated along the light setting dimension. In the experiments described in this paper \( l_i \) denotes quantized light intensity levels. As shown in Fig. 6 for each state \( s_j \), \( U(s, l) \) is estimated as a Gaussian that fits optimally to the existing Q-table values in the row corresponding to \( s_j \). With the utility function \( U(s, l) \), we can estimate the user’s preference level of any pair \((s_j, l_i)\) even if the corresponding Q-table grid is still not explored and contains no entry.

2.4. Optimal control of light setting

Since our goal is to maximally satisfy the user and save energy, the optimization objective function is constructed as follows:

\[
\text{minimize } f_s(l) = U(s, l) + \lambda C(l)
\]

for the current state \( s \). \( C(l) \) is the cost of energy, which is linear in the intensity level: \( C(l) = c_0 l \) where \( c_0 \) is the unit-level cost of energy. Even though \( U(s, l) \) is not convex, the optimization is straight-forward since \( l \) is quantized. A greedy search efficiently yields the optimal light intensity \( l^* \). At the current stage, we do not consider settings based on combination of lights. For each state, the optimal setting for each light is solved separately. In future work, we will introduce more complex ambient settings which include combination of lights.

Figure 6. The utility function is estimated from the Q-table.

Figure 7. The schematic and two views of * lab.

<table>
<thead>
<tr>
<th>cam 1</th>
<th>cam 2</th>
<th>cam 3</th>
<th>cam 4</th>
<th>cam 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>cutting</td>
<td>0</td>
<td>9032</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>scrambling</td>
<td>0</td>
<td>9042</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>eating</td>
<td>0</td>
<td>11397</td>
<td>11414</td>
<td>0</td>
</tr>
<tr>
<td>reading</td>
<td>3082</td>
<td>0</td>
<td>5176</td>
<td>10484</td>
</tr>
<tr>
<td>computer</td>
<td>4640</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>vacuuming</td>
<td>0</td>
<td>9052</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>others</td>
<td>5130</td>
<td>9032</td>
<td>18160</td>
<td>8909</td>
</tr>
</tbody>
</table>

Table 2. Number of frames for each activity from each camera.

3. Experiments

Our testbed, called the * Lab, is a smart studio located at xxx University (Fig. 7). It consists of a living room, kitchen, dining area, and study area. The testbed is equipped with a network of cameras, a large-screen TV, a digital window (projected wall), handheld PDA devices, appliances, and wireless controllers for lights and ambient colors.

3.1. Activity analysis

We acquired approximately 80 minutes of video sequences as the training data for the fine (second)-level of activity classification with spatio-temporal features (Fig. 1). Each sequence is captured from all the 5 installed cameras, and two persons participated in the sequence taking. Performance of the coarse (first)-level activity classification can be found in [13]. Below we only present performance of the second-level activity analysis. The number of frames for each activity in each camera is shown in Table 2. We experimented on the number of K-means clusters (\( N \)) and the episode duration for bag-of-features (\( t \)). A three-fold cross-validation process is used in which when one fold is chosen as test data the other two are chosen as training data. Considering the average precision of all cameras, we observed when \( N > 60 \) and \( t > 12 \) seconds, the performance stays roughly stable. Therefore we chose \( N = 100 \) and \( t = 15 \) seconds. Precision for each activity in each camera can be found in Table 3.
The test data for HRL contains about 45 minutes of video sequences from 4 persons. Each person continuously performs different activities in the environment. At each frame, the best-view camera is chosen for the fine-level activity analysis. The best-view camera is chosen based on the person’s location. The episode is segmented out with a sliding window with duration of 15 seconds and slides every 1 second.

### 3.2. Reinforcement Learning

In order to evaluate the ability of the system to follow the user’s preferences in the light settings, 5 levels of intensities were defined for the 4 lights located at the different areas of the testbed lab. The lights are adjustable using a set of wireless dimmers. In our simulation to study the convergence behavior of the learning system, the mean-square-error (MSE) between the ideal and current compositions of light intensities are given as the feedback to the system. Two volunteer participants executed different activities in different places over different time intervals. Fig. 8 shows the storylines of the examined scenarios. Assuming the participants would not change their behavior, the scenarios were used as daily habits of the users and repeated to construct 10,000 observations for each experience.

At the beginning of each scenario, the system does not have any prior knowledge about the user preferences. (Fig. 9) shows the average penalty in the 1000 past observations versus time. Due to state space abstraction of hierarchical reinforcement learning, the HRL methods converge faster to ideal composition in comparison with the regular RL methods. This means that less time is required by the system to adapt to the user’s ideal light settings. A fast adaptation rate is important when the user occasionally changes his preferred light settings and the system needs a quick adjustment. The length of activities in User 1’s video is less than that of User 2’s. As a result, in the case of User 1 the system does not stay in the same state long enough in each daily episode and the four examined methods perform linearly like each other before they converge.

### 4. Conclusion

We propose to use hierarchical reinforcement learning to adaptively learn user’s preferences through user feedback for the problem of optimal light control. The strength of HRL is that it uses high-level routines to achieve a faster convergence, which is critical for real-time online learning. We presented experiments on sequences taken from...
our testbed environment, both on activity analysis and preference learning. Our work on HRL for online user preference learning is currently focused on defining the routines with a different semantic grouping other than location, to further examine the behavior of HRL in adapting to user preferences. We are also extending the range of services to ambient lighting and background music where the user preferences can span a larger set of states.

References


