Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence

# An Active Learning Approach to Home Heating in the Smart Grid

**Mike Shann** and **Sven Seuken**
Department of Informatics
University of Zurich
{shannm, seuken}@ifi.uzh.ch

## Abstract

A key issue for the realization of the smart grid vision is the implementation of effective demand-side management. One possible approach involves exposing dynamic energy prices to end-users. In this paper, we consider a resulting problem on the user's side: how to adaptively heat a home given dynamic prices. The user faces the challenge of having to react to dynamic prices in real time, trading off his comfort with the costs of heating his home to a certain temperature. We propose an active learning approach to adjust the home temperature in a semi-automatic way. Our algorithm learns the user's preferences over time and automatically adjusts the temperature in real-time as prices change. In addition, the algorithm asks the user for feedback once a day. To find the best query time, the algorithm solves an optimal stopping problem. Via simulations, we show that our algorithm learns users' preferences quickly, and that using the expected utility loss as the query criterion outperforms standard approaches from the active learning literature.

## 1 Introduction

One of society's greatest challenges in the 21st century is the revolution of the energy sector, moving from fossil-based energy sources towards renewable energy like wind and solar. This transition is important to satisfy the growing demand for energy while the annual production of many oil and gas fields is decreasing, and to combat climate change in general and the negative effects of carbon emissions in particular. However, this also creates a number of new challenges for three reasons: energy from renewable sources is very volatile; energy is inherently difficult to store; and the classic model in energy markets is one where supply follows demand. To address these new challenges, governments are investing billions of dollars into the development of the next generation of the electricity grid, the so-called *smart grid* [U. S. Department Of Energy, 2003]. This new electricity network will make it possible to expose real-time prices to end-consumers, use electric vehicles that are plugged into the grid as energy storage devices, and allow power companies to remote control certain home appliances in times when electricity supply is particularly scarce. However, in contrast to the smart grid vision, at the moment most end-users are still facing fixed energy prices or very simple day/night tariffs, and are unaware of changes in the demand or supply of energy.

### 1.1 Demand-Side Management

With renewable energy becoming a larger part of the overall energy mix, it is becoming increasingly difficult for supply to always follow demand. A number of recent economic and technological studies have shown that effective *demand-side management* will be essential for the success of the smart grid [Cramton and Ockenfels, 2011]. This means that in times where energy supply is scarce, the demand for energy must also decrease. One way to achieve this is to expose dynamic energy prices to end-users in real time such that they can adjust their demand accordingly. At the moment, the biggest demand-response effects come from big companies who already face dynamic prices and can shift some of their energy usage [VDE, 2012]. However, in the future, the percentage of electricity consumed by end-users will increase because more and more cars will be electric vehicles, and an increasing number of homes will use electric heat pumps and air conditioners. Even if just part of the population adopts energy tariffs with dynamic prices, effective demand-response management for end-users will become an important challenge.

### 1.2 Home Heating with Smart Thermostats

In this paper, we focus on one particular facet of demand-response management: the problem of adaptively heating (and cooling) a user's home given dynamic electricity prices. This addresses an important problem because cooling and heating accounts for the largest part of end-users' energy bills. We consider a future smart grid design, where at least some end-consumers are exposed to dynamic energy prices. To optimize their utility, those users will have to react to dynamic prices in real-time, trading off their comfort (at different temperature levels) with the costs for heating or cooling. Obviously, it is infeasible for a user to always manually change the temperature when a price change occurs. Instead, we envision *smart thermostats* that will automatically reduce the energy consumption of the house when prices are high, but only as much as is justified by the cost savings.

Designing a smart thermostat is a difficult problem because automatically adjusting the temperature requires know-

ing how the user trades off comfort for money. Some users may have a high value for comfort and may be willing to pay a lot for a perfectly-heated home. Others may be relatively insensitive to temperature changes, and instead would prefer to save on energy costs. Because of this user heterogeneity, the smart thermostat needs to *elicit* the user's preferences and learn this trade-off over time, which makes this a formidable AI problem in the computational sustainability domain.

Yet, even the most sophisticated thermostats currently on the market do not consider this trade-off. The existing devices are able to monitor a home's energy usage and suggest energy saving measures (e.g., *Alert Me*), or they can learn a user's daily schedule and adjust the times at which the house is heated or cooled accordingly (e.g., *Eco Factor* and *Nest*). However, these devices are completely unresponsive to energy price changes. Recent academic work on adaptive home heating has focused on learning the thermal properties of a house, but has also not considered how the user trades off between comfort and money [Rogers *et al.*, 2011]. Naturally, end-consumers are currently still very sceptical regarding the benefits of the smart grid [Jung, 2010]. Many believe that their comfort levels will be reduced and that they will only save little if any money. We argue that a smart thermostat that automatically reacts to price changes is necessary to realize demand-response management, and would also be in the interest of end-users. However, it must be non-intrusive and simple to use, for end-consumers to adopt this technology.

### 1.3 Overview of Contributions

The main contribution of this paper is an active learning algorithm for the adaptive home heating problem. Our algorithm uses Bayesian inference to learn the user's preferences over time, automatically adjusts the temperature as prices change, and requests new feedback from the user, but only once a day. We explicitly model the user's comfort-cost trade-off by separating the user's value function (for temperature) from the cost function (for heating or cooling). We propose an algorithm that involves solving an optimal stopping problem to find the optimal time to query the user. We evaluate our algorithm in an online fashion via simulations. We find that using the user's expected utility loss as the query criterion outperforms standard approaches from the active learning literature. To the best of our knowledge, we are the first to propose an active learning approach to address demand-side management in the smart grid.

## 2 Related Work

**Automated Control in the Smart Grid.** Ramchurn et al. [2012] provide a good introduction to smart grids and the demand-response management challenge. Rogers et al. [2011] study the adaptive home heating problem. However, their focus is on learning the thermal properties of a house and predicting environmental parameters, to optimize the heating schedule. They assume that the user's preferred temperature is known in advance and do not consider the comfort-cost trade-off. McLaughlin et al. [2012] consider the same problem but also assume that the user's desired temperature is known to the algorithm. Vytelingum et al. [2010] study micro-storage management for the smart grid, and devise agent strategies that automatically react to price changes. However, they assume that the amount of energy each user desires per time period is known in advance, and thus the problem of eliciting users' preferences also does not arise in their model. Finally, Jia et al. [2012] consider the retailer's perspective, and provide a solution for optimal pricing of energy, given that users trade off comfort for cost. However, they also do not consider how a demand-response system would learn about a user's trade-off preferences. Overall, our literature review suggests that the problem of *eliciting* and *learning* user preferences in the smart grid has largely been ignored by the research community so far.

**Preference Elicitation and Active Learning.** Our work primarily uses techniques from preference elicitation [Boutilier, 2002] and active learning [Settles, 2009]. Our Bayesian inference algorithm is inspired by the preference elicitation approach by Chajewska et al. [2000], who use the expected value of information as their query criterion. However, while they consider a domain where arbitrary queries can be synthesized, we consider the problem of selecting the best query from a stream of potential queries which is called *selective sampling* or *stream-based sampling*. Cesa-Bianchi et al. [2006] and Beygelzimer et al. [2009] propose randomized selective sampling algorithms that have good convergence guarantees in the limit, but do not aim to optimize each individual sample. Our query technique is more similar to the approach used by Cohn et al. [1996], in that we aim to minimize the learner's expected error with every individual query. Our work is also related to the label efficient prediction algorithms by Helmbold et al. [1997] and Cesa-Bianchi et al. [2005]. Their algorithms handle the restriction that the learner can only ask a limited number of times, however, they cannot handle context variables, like price for example. In contrast, Krause and Ong [2011] present bandit algorithms that explicitly take context into account. However, they assume that the algorithm receives feedback about the user's utility in every time step which is not given in our domain. Finally, our problem can also be framed as a *partial monitoring game with side information* [Cesa-Bianchi and Lugosi, 2006]. However, existing algorithms for this framework operate in a prior-free domain [Bartók and Szepesvári, 2012], while we assume a Bayesian learning framework.

## 3 The Model

### 3.1 Problem Statement

We consider the problem of adaptive home heating over a horizon of $N$ days, where each day consists of $K$ time steps. The price for energy is modeled using a discrete Markov process $\{p_t : t \leq KN\}$. We use $T_{out}$ to denote the current outside temperature, and $T$ to denote the current temperature inside the house. The user's utility is separated into two components. First, it depends on his comfort level, which is mainly determined by the inside temperature $T$ but also influenced by the outside temperature $T_{out}$. Second, the utility depends on the cost the user has to pay for heating the house, which is a function of the desired inside temperature $T$, the outside temperature $T_{out}$, and most importantly the current price for energy $p_t$. We denote the user's utility by $u(p_t, T, T_{out})$.

Our goal is to design an active learning algorithm that learns the user's preferences over time and controls the house's temperature in a semi-automated way. Every time step, the algorithm receives as input the current price $p_t$. At most once per day, the algorithm can query the user for the temperature that is currently *optimal* for him:

$$T_{opt}(p_t, T_{out}) = \arg\max_T u(p_t, T, T_{out}). \quad (1)$$

We assume that if the algorithm decides to issue a query, the user provides a temperature value which the algorithm uses to update its model of the user's preferences. Based on its current knowledge, the algorithm then sets the temperature to its current best *estimate* of the optimal temperature, which we denote by $\widehat{T}_{opt}(p_t, T_{out})$.[1] Note that we often use $T_{opt}$ and $\widehat{T}_{opt}$ without the parameters $p_t$ and $T_{out}$ to simplify notation. Our goal is to minimize the user's cumulative utility loss:

$$L = \sum_{t=1}^{KN} \Big( u(p_t, T_{opt}, T_{out}) - u(p_t, \widehat{T}_{opt}, T_{out}) \Big). \quad (2)$$

The one-query-per-day restriction is motivated by our goal of designing a non-intrusive smart thermostat that end-consumers are willing to use. Of course, many other design choices regarding the interaction mode are conceivable, including several queries per day, queries at a fixed time (e.g. in the evening), or even a more intense preference elicitation phase at the beginning of the learning process.

### 3.2 The User's Utility Function

Inherent to the home heating problem is the user's trade-off between comfort and cost. To model this, we assume a value function $v(T, T_{out})$ that quantifies (in currency) the user's level of comfort for temperature $T$ given $T_{out}$, and a cost function $c(p_t, T, T_{out})$ that quantifies how expensive it is to heat the house to temperature $T$ at current price $p_t$ given $T_{out}$. The user's utility is the difference between value and costs:

$$u(p_t, T, T_{out}) = v(T, T_{out}) - c(p_t, T, T_{out}). \quad (3)$$

**Value Function.** Prior research on thermal comfort has shown that the colder it is outside, the lower the user's acceptable indoor temperature [Peeters *et al.*, 2009]. This suggests that the user's most preferred temperature also depends on the current outside temperature. Formally, we let $T^*$ denote the user's preferred temperature at $T_{out} = 0$, and we let $m$ denote the slope with which the preferred temperature increases as the outside temperature increases. We denote the user's preferred temperature by $T_{pref}(T_{out}) = T^* + mT_{out}$.

Following prior work on home heating (e.g., [Rogers *et al.*, 2011]), we assume that the user incurs a utility loss if the inside temperature deviates from his preferred temperature. In particular, we assume that the utility loss is quadratic in $(T_{pref} - T)$, i.e., in the difference between the user's preferred temperature and the actual inside temperature.

[1] Note that $\widehat{T}_{opt}$ may be different from the temperature value provided by the user, which is consistent with our Bayesian approach, but may be confusing for the user in practice. Of course, to improve usability, the smart thermostat could also "ignore" the Bayesian model for one time step, and simply set the temperature to the value provided by the user.

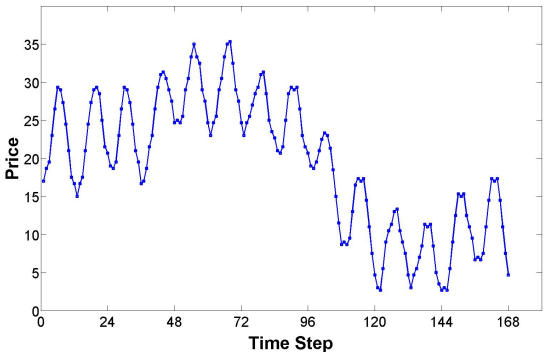


Figure 1: An illustration of the stochastic price process, here over 7 days. The price process has two periodic peaks per day with random fluctuations that follow a random walk.

Peeters et al. [2009] have shown that people are more sensitive to temperature deviations the colder it is outside. To model this, we use an exponential function parameterized by $b$, which denotes the user's sensitivity if $T_{out} = 0$, and $c$, which determines how much the user's sensitivity changes as the outside temperature changes. Finally, we let $a$ denote the user's value for his most preferred temperature (i.e., when $T^* + mT_{out} = T$). Putting all of this together, we arrive at the following value function formulation:

$$v(T, T_{out}) = a - \underbrace{b \cdot e^{-c \cdot T_{out}}}_{\text{sensitivity}} \big( \underbrace{(T^* + mT_{out})}_{\text{preferred temp.}} - T \big)^2 \quad (4)$$

**Cost Function.** The user's cost function is given by the following equation:

$$c(p, T, T_{out}) = p|T - T_{out}|. \quad (5)$$

This function captures the fact that the flow of heat between a building's interior and exterior is proportional to the temperature difference, which implies that the amount of energy necessary to heat a house also depends on the temperature difference. Note that this function correctly models heating and cooling, since it only depends on the temperature difference.

Combining the value and the cost function, we obtain the following *linearly separable utility function*:

$$u(p, T, T_{out}) = \underbrace{a - be^{-cT_{out}} \big( (T^* + mT_{out}) - T \big)^2}_{\text{value}} - \underbrace{p|T - T_{out}|}_{\text{cost}}$$

### 3.3 The Stochastic Price Process

Because the algorithm only queries the user once per day, we are interested in the daily price dynamics. An important feature of the daily energy prices are two peaks, one in the morning at around 8 a.m., and one in the evening at around 6 p.m. We model this periodicity using a sine function, following Weron [2006]. To model any random price movements (e.g., due to demand or supply changes) we use a discrete symmetric random walk. Put together, the price process is given by:

$$p_t = A \sin(\omega t + \phi) + B + p_{t-1} + X_t, \quad (6)$$

where $A$ is the amplitude of the sine, $\omega$ is the periodicity, $\phi$ is the phase shift, $B$ is the offset, and $X_t$ is a Bernoulli variable corresponding to the random walk. We use the notation $\mathrm{P}(p_{t'}|p_t)$ to denote the conditional probability of encountering the price $p_{t'}$ given $p_t$. See Figure 1 for an illustration of the price process over 7 days with 24 time steps per day.

## 4 The Active Learning Algorithm

The active learning algorithm we propose consists of two main components: 1) a Bayesian learning component that learns the parameters of the user's utility function over time, and 2) a query component that decides when to ask the user for new feedback (once per day). In Section 4.1, we describe the high-level algorithmic framework, before diving into the details of the two components in the following sections.

### 4.1 The Algorithmic Framework

Every day, the algorithm's goal is to select the best query from the stream of prices it encounters. Loosely speaking, it faces the following gamble. Either sample at the current price or wait and hope that a future query will yield a more useful sample. We will use the notion of a *gain function* $G(p_t)$ to measure the "usefulness" of a query at price $p_t$. It is intuitive, for example, that querying at a price at which the user has already given feedback before is less useful than asking at a price that has not been encountered before. The different gain functions we consider (information gain and variance reduction) measure usefulness in different ways and thus lead to different decisions regarding the optimal query time.

Given the $K$ times steps per day, the algorithm's goal is to find the optimal stopping time $t^* \in \{1, \ldots, K\}$ at which the expected gain $G$ of a query is highest:

$$t^* = \arg\max_t \mathrm{E}[\mathrm{G}(p_t)]. \tag{7}$$

To find the optimal stopping time, the algorithm computes an *optimal stopping policy* $\pi(t, p_t) \rightarrow \{sample, continue\}$. For each time $t$ and price $p_t$, this policy prescribes whether to ask the user for feedback now, or whether to wait. This policy can be computed by dynamic programming [Peskir and Shiryaev, 2006]. Keep in mind that the algorithm computes a new optimal stopping policy at the beginning of every day.

If the algorithm decides to request feedback, it asks the user what his preferred temperature is right now given $p_t$ and $T_{out}$. The user decides how to trade off his comfort level against the costs for heating, and then provides a temperature value $y_t$ to the algorithm. Using this new data point, the algorithm updates its model of the user's utility function using Bayes' rule. Finally, the algorithm sets the optimal temperature, $\widehat{T}_{opt}$, taking into account its prior knowledge and all feedback data it has gathered about the user's preferences so far. The user then suffers a utility loss of $u(p_t, T_{opt}, T_{out}) - u(p_t, \widehat{T}_{opt}, T_{out})$, which is not observed by the algorithm, but which we use to measure the performance in our simulation in Section 5. The whole active learning framework is shown in Algorithm 1.

**Algorithm 1:** Active Learning Framework

**Input**: prior $(m_{\boldsymbol{\theta}}, \Sigma_{\boldsymbol{\theta}})$; noise variance $\sigma_n^2$
**Variables**: current price $p_t$, optimal stopping policy $\pi$
**begin**
  **for** *d=1 to # of days* **do**
    **for** *t=1 to # of time steps per day* **do**
      $p_t \leftarrow getNextPrice(p_{t-1})$
      **if** *t =1* **then**
        //allow a new query
        canAsk $\leftarrow$ **true**
        $\pi \leftarrow OptimalStopping(p_t)$
      **if** *canAsk* **then**
        //decide whether to query user
        **if** $\pi(p_t, t) = sample$ **then**
          $y_t \leftarrow getUserFeedback()$
          $BayesianUpdate(p_t, y_t)$
          canAsk $\leftarrow$ **false**
      $\widehat{T}_{opt} \leftarrow SetTemperature(p_t, T_{out})$

### 4.2 Bayesian Updating & Setting the Temperature

Recall that the user's optimal temperature is given by:

$$T_{opt}(p_t, T_{out}) = \arg\max_T u(p_t, T, T_{out}). \tag{8}$$

Based on the functional form of the user's utility function described in Section 3.2, we can calculate the first order condition and solve for $T$, and arrive at the following equation for the user's optimal temperature:

$$T_{opt}(p, T_{out}) = T^* + mT_{out} \pm p\frac{e^{cT_{out}}}{2b} \tag{9}$$

Note that $a$ does not matter for the optimization, and we only have to learn the parameter vector $\boldsymbol{\theta} = (b, c, T^*, m)$.

**Bayesian Updating.** We assume that the parameter vector $\boldsymbol{\theta}$ is normally distributed, and therefore define a Gaussian prior $\mathrm{P}(\boldsymbol{\theta}) = \mathcal{N}(m_{\boldsymbol{\theta}}, \Sigma_{\boldsymbol{\theta}})$. Furthermore, we assume that the user makes mistakes when giving feedback $y_t$ to the thermostat. We model this with i.i.d. additive Gaussian noise, $y_t = T_{opt} + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$ is a normally distributed random variable with mean 0 and *noise variance* $\sigma_n^2$. Thus, the likelihood of $y_t$ is also normally distributed with mean $T_{opt}$ and variance $\sigma_n^2$:

$$\mathrm{P}(y_t|p_t, \boldsymbol{\theta}) \propto \exp\Big(-\frac{1}{2\sigma_n^2}(y_t - T_{opt})^2\Big). \tag{10}$$

The posterior is then computed as the product of the prior and the likelihood according to Bayes' rule:

$$\mathrm{P}(\boldsymbol{\theta}|p_t, y_t) \propto \mathrm{P}(\boldsymbol{\theta}) \cdot \mathrm{P}(y_t|p_t, \boldsymbol{\theta}). \tag{11}$$

To update the posterior distribution after a sample point $(p_t, y_t)$ has been gathered, we use Eq. (11) recursively, using the posterior after $k{-}1$ observations as the prior for the $k^{th}$ update step:

$$\mathrm{P}(\boldsymbol{\theta}|D_{k-1} \cup (p_t, y_t)) \propto \mathrm{P}(\boldsymbol{\theta}|D_{k-1}) \cdot \mathrm{P}(y_t|p_t, \boldsymbol{\theta}), \tag{12}$$

where $D_{k-1} = \{(p_{i_1}, y_{i_1}), ..., (p_{i_{k-1}}, y_{i_{k-1}})\}$ denotes all $k{-}1$ data points the algorithm has gathered until time step $t-1$.

**Setting the Temperature.** Finally, the thermostat sets the estimated optimal temperature (according to its model of the user's preferences) by computing the expected value of $T_{opt}$, weighting each of the possible values for the parameters $\boldsymbol{\theta}$ by their posterior probability:

$$\widehat{T}_{opt}(p_t, T_{out}) = \mathbb{E}_{\boldsymbol{\theta}}[T_{opt}] \tag{13}$$

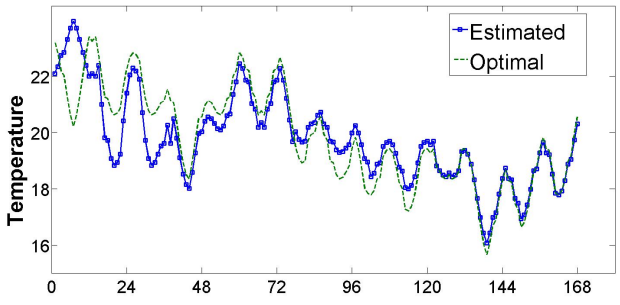


Figure 2: A sample run, illustrating how our algorithm learns the user's preferences over time (here 7 days).

Figure 2 illustrates what our algorithm does in practice. The figure shows a sample run from our simulation (described below), over 7 days, here with 24 time steps per day. The blue line represents the user's true optimal temperature. The green line represents the estimated temperature values that our algorithm sets based on its user model. As one can see, although the estimated temperature is initially off by 2 to 3 degrees, it quickly converges to the true optimal temperature.

### 4.3 Optimal Stopping using Information Gain

Now that we have introduced the learning component of our algorithm, we move on to the description of the query component. First, we formalize the optimal stopping problem and show how to solve it. Then we introduce *information gain* as the first gain function, or query criterion. In the next two sections, we refine those initial approaches, leading to an improved version of the optimal stopping algorithm as well as to more sophisticated query criteria.

**Computing the Optimal Stopping Policy.** Recall from Section 4.1 that the optimal stopping policy is a function $\pi(t, p)$ that for every price $p$ and time step $t$ prescribes whether to query now, or whether to wait. Obviously, the policy only prescribes to *wait* if the immediate gain from querying the user now is lower than the expected future gain from waiting and querying later.

While $G(p_t)$ denotes the immediate gain from querying now at price $p_t$, we let $S_t$ denote the expected gain at time step $t$ when following the optimal stopping policy at every time step going forward from $t$. $S_t$ is defined recursively as:

$$S_t = G(p_t) \quad \text{for } t = K \quad \text{(last time step)}$$
$$S_t = \max\{G(p_t), \mathbb{E}[S_{t+1}|p_t]\} \quad \text{for } t = K{-}1, \ldots, 1. \quad (14)$$

To derive the optimal policy, we compare the gains $G(p_t)$ at time step $t = 1, \ldots, K{-}1$ to the expected future gains $\mathbb{E}[S_{t+1}|p_t]$ for all possible prices $p_t$. If $G(p_t) \geq \mathbb{E}[S_{t+1}|p_t]$, then the optimal policy states that we should query, i.e., $\pi(t, p_t) = sample$. Otherwise, $\pi(t, p_t) = continue$.

Note that the first price $p_1$ is known, and thus all future prices $p_t$ that could possibly be encountered until the end of the day can be computed by adding or subtracting a) the random walk price increment per time step, and b) the price movements according to the daily price process model.

**Algorithm 2:** Computing the Optimal Stopping Policy

**Input**: starting price $p_1$
**Output**: optimal stopping policy $\pi$
**begin**
  $S \leftarrow 0$
  **for** $t =$ *# of time steps per day to 1* **do**
    **forall the** *reachable prices* $p$ **do**
      **if** *t = # of time steps per day* **then**
        $\pi(t, p) = sample$
      **else**
        **if** *t = # of time steps per day* $-1$ **then**
          $S_{t,p} \leftarrow \frac{1}{2}[\mathrm{G}(p+1) + \mathrm{G}(p-1)]$
        **else**
          $S_{t,p} \leftarrow \frac{1}{2}[\max\{\mathrm{G}(p+1), S_{t+1,p+1}\} + \max\{\mathrm{G}(p-1), S_{t+1,p-1}\}]$
        **if** $G(p) \geq S_{t,p}$ **then**
          $\pi(t, p) = sample$
        **else**
          $\pi(t, p) = continue$
  **return** $\pi$

Algorithm 2 shows how the optimal stopping policy is computed for all time steps and all possible prices. To simplify the exposition of the algorithm, we assume here that the price process is a symmetric random walk with step size 1. However, it is straightforward to adopt the algorithm to more complicated price processes such as the one defined in Section 3.3. We use the variable $S_{t,p}$ to denote the expected gain at time step $t$, given price $p$, i.e. $S_{t,p} = \mathrm{E}[S_t|p]$.

**Query criterion: Information Gain.** So far, we have left the gain function $G(p_t)$ unspecified. However, to instantiate the optimal stopping algorithm, we need to define one particular gain function, or query criterion, $\mathrm{G}(p_t)$, that quantifies how useful a query is at a price $p_t$ (note that we use the terms *gain function* and *query criterion* interchangeably). The first criterion we discuss is *information gain* which measures how much the uncertainty about the parameters $\boldsymbol{\theta}$ is reduced by adding an observation $y_t$ [Cover and Thomas, 2006]. This is expressed using the mutual information $I(\theta, y_t) = H(\theta) - H(\theta|y_t)$, where $H(\cdot)$ is the differential entropy [Cover and Thomas, 2006]. Intuitively, the higher the uncertainty (or variance) of $T_{opt}$ at a given price, the more information can be gathered by querying at this price. It can be shown that the information gain for a given price is equivalent to the variance of the predicted optimal temperature $T_{opt}$ [MacKay, 1992]. Thus, we define our first query criterion as:

$$\mathrm{G}^{inf}(p_t) = Var[T_{opt}(p_t)] \quad (15)$$

Note that the user's utility actually also depends on the outside temperature $T_{out}$. However, in this paper, we do not assume that the algorithm has a model for $T_{out}$. Thus, our formulation of the optimal stopping problem is only optimal with respect to the stochastic price process and implicitly assumes a fixed value for $T_{out}$. But it is straightforward to extend the algorithm by incorporating a model for $T_{out}$ as well.

### 4.4 Optimal Stopping using Temperature Loss

Note that the basic version of the optimal stopping algorithm neglects the fact that until the algorithm asks the user for feedback, the user has already incurred a utility loss every time step. Therefore, we now re-formulate the optimal stopping problem using *loss functions*, with the new goal of minimizing the expected total loss. Therefore, we define our gain function $G(p_t)$ to be a loss function multiplied by $-1$, i.e., $G(p_t) = -L(p_t)$, such that minimizing the expected loss is equivalent to maximizing the expected gain.

We define the function $L^{now}(p_t)$ that measures the loss the user incurs at time $t$ given price $p_t$ if the algorithm estimates the optimal temperature with its current knowledge without issuing a query. Thus, the algorithm will incur loss $L^{now}(p_t)$ at every time step $t$ until it decides to query the user. However, if the algorithm decides to issue a query at time $t$, then the loss incurred will be smaller than $L^{now}(p_t)$ because the algorithm will be able to estimate the temperature more accurately due to one additional data point. This leads to the following new definition of $S_t$:

$$S_t = G(p_t) \quad \text{for } t = K \quad \text{(last time step)}$$
$$S_t = \max\{G(p_t), -L^{now}(p_t) + \mathrm{E}[S_{t+1}|p_t]\} \text{ for } t = K\text{–}1,...,1.$$

The term $-L^{now}(p_t)$ in the last equation reflects the loss that the user incurs if the algorithm does not issue a query at time $t$, while the (smaller) loss incurred if the algorithm issues a query will be incorporated in the gain function $G(p_t)$, which we define in the next section. As before, the optimal stopping policy can be computed using the approach summarized in Algorithm 2, but adapting the equations for the expected future gains $S_{p,t}$ according to the new formulation.

**Query criterion: Temperature Loss.** To instantiate the new *loss-based* optimal stopping algorithm, we follow an idea from [Cohn *et al.*, 1996], and specify as our new goal to select the query that minimizes the expected squared error in the temperature estimation. This is motivated by the fact that the expected squared error of a learner can be decomposed into squared bias and variance, the so-called *bias-variance decomposition* [Geman *et al.*, 1992], which states that we can approximate the expected squared predictive error if the bias of the learner is sufficiently small compared to the variance.

First, let us revisit $L^{now}(p_t)$. Due to the bias-variance decomposition, we can approximate this function using the variance of the predicted temperature: $L^{now}(p_t) = Var[T_{opt}(p_t)]$. To obtain a gain function $G(p_t)$, we need the expected (posterior) variance of $T_{opt}$, condition on sampling at a given price $p_t$. We let $L^{ask,t}_{temp}(p)$ denote the expected conditional variance of $T_{opt}$ at price $p$, if the user was queried at time step $t$, i.e. $L^{ask,t}_{temp}(p) = Var[T_{opt}(p)|(p_t, y_t)]$. The gain function that we define now amounts to quantifying the expected predictive loss until the end of the day plus the expected loss of one additional day, given the user was queried. Adding the expected loss of one additional day is a heuristic to account for the future differences in losses due to the particular query. This is only a heuristic as it does not account for *all* effects on losses in future days, because it ignores the fact that the algorithm will be able to issue a new query on the next day (and on every day thereafter).[2] The query criterion is then defined as follows:[3]

$$G^{loss\_temp}(p_t) =$$
$$-\Big(L^{ask,t}_{temp}(p_t) + \underbrace{\sum_{t'=t+1}^{K} \mathrm{E}[L^{ask,t}_{temp}(p_{t'})|p_t]}_{\text{loss until end of day}} + \underbrace{\sum_{t'=K+1}^{2K} \mathrm{E}[L^{ask,t}_{temp}(p_{t'})|p_t]}_{\text{loss next day}}\Big)$$

Note that $\mathrm{E}[L^{ask,t}_{temp}(p_{t+i})|p_t]$ denotes the expectation of $L^{ask,t}_{temp}(p_{t+i})$ with respect to the condition probability distribution given by the price process, i.e., according to $\mathrm{P}(p_{t+i}|p_t)$, as defined in Section 3.3.

### 4.5 Optimal Stopping using Utility Loss

The query criterion we develop in this section is based on the following insight: minimizing the expected squared error of the temperature estimation (as we did in the previous section) misses the fact that the user primarily cares about his *utility losses*, and that an error in the temperature estimation can only be a proxy for that. Thus, our new goal is to directly minimize the user's expected utility loss.

Analogously to the temperature variance criterion, we can approximate the user's squared utility loss with the variance of the utility function. To arrive at the expected utility loss we can simply take the square root of the variance of the utility. Therefore, define $L^{now}_u(p) = \sqrt{Var[u(p)]}$, and similarly $L^{ask,t}_u(p) = \sqrt{Var[u(p)|(p_t, y_t)]}$. The following query criterion minimizes the expected square root of the variance of the utility function, which is equivalent to choosing a sample point that minimizes the user's expected utility loss:

$$G^{loss\_util}(p_t) =$$
$$-\Big(L^{ask,t}_{u}(p_t) + \underbrace{\sum_{t'=t+1}^{K} \mathrm{E}[L^{ask,t}_{u}(p_{t'})|p_t]}_{\text{loss until end of day}} + \underbrace{\sum_{t'=K+1}^{2K} \mathrm{E}[L^{ask,t}_{u}(p_{t'})|p_t]}_{\text{loss next day}}\Big)$$

This query criterion together with the optimal stopping formulation described above is the ultimate query component that we propose for our active learning algorithm.

## 5 Experiments

We evaluate our active learning approach via simulations, following the basic structure of Algorithm 1. For the learning and prediction part of the algorithm, we perform a non-linear regression using a Bayesian linear parameter model.

### 5.1 Bayesian Linear Parameter Model

Recall that the optimal temperature is a non-linear function of the input variables $p_t$ and $T_{out}$. However, if we fix the parameter $c$, we can write the optimal temperature as a linear parameter model:

$$T_{opt}(p_t, T_{out}) = w_0 + w_1 T_{out} + w_2 p \cdot e^{cT_{out}}/2 \qquad (16)$$

[2] Note that solving an optimal stopping problem over a horizon of $N$ days with $K$ time steps each quickly becomes computationally infeasible, even for moderate values of $N$ and $K$.

[3] To simplify the notation for the summation indices, we only state the criterion for the first day.

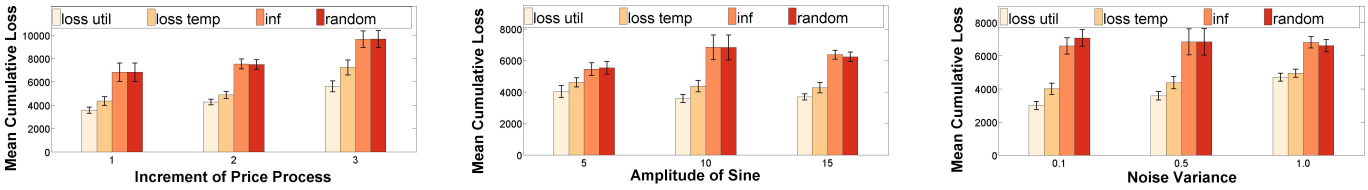


Figure 3: Simulation results comparing four different query criteria: (a) varying the increment of the price process; (b) varying the amplitude of the sine of the price process; (c) varying the noise variance $\sigma_n^2$.

We can identify the weights as follows: $w_0 = T^*$, $w_1 = m$ and $w_2 = 1/b$. We augment the input vector with an offset, such that $\mathbf{x} = (1, p_t, T_{out})$ and write

$$T_{opt}(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}), \tag{17}$$

where $\mathbf{w}=(w_0, w_1, w_2)^T$ and $\boldsymbol{\phi}(\mathbf{x})=(1, T_{out}, pe^{cT_{out}}/2)^T$. Due to our assumption of a Gaussian prior and a Gaussian additive noise model, the posterior probabilities are likewise Gaussian and we can perform Bayesian regression using the Bayesian linear parameter model [Bishop, 2006].

### 5.2 Experimental Set-up

For all experiments, we use the following basic set-up. We use $N = 30$ days, each day consisting of $K = 12$ time steps. The prior means are 22 for $w_0$ (i.e. $T^*$), 0.1 for $w_1$ (i.e. $m$), and 0.2 for $w_2$ (i.e., $1/b$). The values $T^* = 22$ and $m = 0.1$ are similar to the values reported by Peeters et al. [2009]. The prior variances are fixed as $\boldsymbol{\sigma^2}=(1, 0.1, 0.1)$. The noise variance, which describes the user's ability to provide accurate temperature values (see also Eq. (10)), is set to $\sigma_n^2 = 0.5$.

For the sine of the price process, we set the amplitude $A = 10$, the offset $B = 20$, the periodicity $\omega = 4\pi/K$, and the phase shift $\phi = 4\pi/3$. The increment of the random walk is 1, i.e. $X_t \in \{-1, 1\}$. The daily variations of the outside temperature are modeled using a sine function with offset 5 and amplitude 5. Thus, $T_{out}$ ranges from 0 to 10 degrees during a day, which are typical heating conditions [Peeters *et al.*, 2009]. The parameter $c$ is set to 0.01. We also conducted the simulations with higher values of $c$ but found qualitatively similar results. Each experiment is repeated for 100 trials, and in every trial, a user type is drawn randomly from the Gaussian prior distribution.

### 5.3 Results

We compare the performance of the following four query criteria: (1) $G^{inf}$, (2) $G^{loss\_temp}$, (3) $G^{loss\_util}$, and (4) random querying. All four query criteria are run in parallel, which implies that they see the same price process and even get the same samples if they perform a query at the same time step.

We vary the parameters that we identified to have a significant impact on the performance of the query criteria. Figure 3(a) shows the results of increasing the increment of the random walk, $X_t$, from 1 to 2 to 3. As one can see, the query criterion $G^{loss\_util}$ performs significantly better than all other criteria, for small as well as for large price increments. In Figure 3(b), we present performance results varying the amplitude of the sine of the prices process from 5 to 10 to 15. Again, $G^{loss\_util}$ outperforms all other query criteria for all three settings. Lastly, in Figure 3(c), we vary the noise variance, $\sigma_n^2$, from 0.1 to 0.5 and 1.0. Here, $G^{loss\_util}$ performs significantly better than all query criteria for $\sigma_n^2 = 0.1$ and $\sigma_n^2$ it performs equally well as $G^{loss\_temp}$ for $\sigma_n^2 = 1.0$. In summary, $G^{loss\_util}$ is never worse than the other criteria, and in most settings significantly outperforms all other criteria.

The results also demonstrate that the information gain criterion, i.e., $G^{inf}$, performs much worse than $G^{loss\_temp}$ and $G^{loss\_util}$. This is mainly due to the fact that the latter two criteria take the loss over the whole day into account, whereas information gain neglects this. A second finding is that the larger the noise, the smaller the differences between the individual criteria. This also makes sense, because lots of noise decreases the predictability of the queries which decreases the value of sophisticated optimized techniques.

## 6 Conclusion

In this paper, we have studied the problem of adaptively heating a home given dynamic energy prices. We have presented a novel active learning algorithm that determines the optimal time to query the user for feedback, learns the user's preferences via Bayesian updating, and automatically sets the temperature on the user's behalf as prices change. Given the constraint of at most one query per day, determining the optimal query time requires solving an optimal stopping problem. Via simulations, we have demonstrated that a query criterion that minimizes the user's expected utility loss outperforms standard approaches from the active learning literature.

It is important to note that we have purposefully presented a relatively simple user model and made a number of simplifying assumptions that we will relax in future work. As a first step, we plan on incorporating the temporal dynamics of heating as well as weather forecasts into our model. This will give rise to a sequential planning problem, which we can combine with our active learning algorithm.

We believe that AI techniques such as preference elicitation and active learning are essential to mediate the interactions between end-consumers and the energy market. To realize the smart grid vision of the future, the design of suitable user interfaces and the use of learning algorithms may ultimately prove to be as important as the economic design of the energy market or the technical aspects of the smart grid.

## Acknowledgments

We would like to thank Timo Mennle and Siddhartha Ghosh for insightful discussions and the anonymous reviewers for their helpful comments.

## References


[Bartók and Szepesvári, 2012] Gábor Bartók and Csaba Szepesvári. Partial Monitoring with Side Information. In *Proceedings of Algorithmic Learning Theory (ALT)*, 2012.

[Beygelzimer *et al.*, 2009] Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. Importance Weighted Active Learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2009.

[Bishop, 2006] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[Boutilier, 2002] C. Boutilier. A POMDP Formulation of Preference Alicitation Problems. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2002.

[Cesa-Bianchi and Lugosi, 2006] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[Cesa-Bianchi *et al.*, 2005] Nicolò Cesa-Bianchi, Gábor Lugosi, and Gilles Stoltz. Minimizing Regret with Label Efficient Prediction. *IEEE Transactions of Information Theory*, 51(6):2152–2162, 2005.

[Cesa-Bianchi *et al.*, 2006] Nicolò Cesa-Bianchi, Claudio Gentile, and Luca Zaniboni. Worst-case Analysis of Selective Sampling for Linear Classification. *Journal of Machine Learning Research*, 7:1205–1230, 2006.

[Chajewska *et al.*, 2000] U. Chajewska, D. Koller, and R. Parr. Making Rational Decisions Using Adaptive Utility Elicitation. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2000.

[Cohn *et al.*, 1996] David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. Active Learning with Statistical Models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

[Cover and Thomas, 2006] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (2. ed.)*. Wiley, 2006.

[Cramton and Ockenfels, 2011] Peter Cramton and Axel Ockenfels. Economics and Design of Capacity Markets for the Power Sector. Working Paper, University of Maryland, May 2011.

[Geman *et al.*, 1992] Stuart Geman, Elie Bienenstock, and René Doursat. Neural Networks and the Bias/Variance Dilemma. *Neural Computation*, 4(1):1–58, January 1992.

[Helmbold and Panizza, 1997] David P. Helmbold and Sandra Panizza. Some Label Efficient Learning Results. In *Proceedings of the Conference on Learning Theory (COLT)*, 1997.

[Jia and Tong, 2012] Liyan Jia and Lang Tong. Optimal Pricing for Residential Demand Response: A Stochastic Optimization Approach. In *Proceedings of the Allerton Conference on Communication, Control and Computing*, 2012.

[Jung, 2010] Alexander Jung. Teure Ersparnis. http://www.spiegel.de/spiegel/0,1518,711967,00.html. Accessed: October 2010.

[Krause and Ong, 2011] Andreas Krause and Cheng Soon Ong. Contextual Gaussian Process Bandit Optimization. In *Proceedings Neural Information Processing Systems (NIPS)*, 2011.

[MacKay, 1992] David J. C. MacKay. Information-based Objective Functions for Active Data Selection. *Neural Computation*, 4(4):590–604, 1992.

[McLaughlin *et al.*, 2012] Zhe Yu Linda McLaughlin, Liyan Jia, Mary C. Murphy-Hoye, Annabelle Pratt, and Lang Tong. Modeling and Stochastic Control for Home Energy Management. In *Proceedings of the Power and Energy Society (PES) General Meeting*, 2012.

[Peeters *et al.*, 2009] Leen Peeters, Richard de Dear, Jan Hensen, and D'haeseleer William. Thermal Comfort in Residential Buildings: Comfort Values and Scales for Building Energy Simulation. *Applied Energy*, 86(5):772–780, 2009.

[Peskir and Shiryaev, 2006] Goran Peskir and Albert Shiryaev. *Optimal Stopping and Free-Boundary Problems*. Lectures in Mathematics, ETH Zürich. Birkhäuser Verlag, 2006.

[Ramchurn *et al.*, 2012] Sarvapali Ramchurn, Perukrishnen Vytelingum, Alex Rogers, and Nicholas R. Jennings. Putting the "Smarts" into the Smart Grid: A Grand Challenge for Artificial Intelligence. *Communications of the ACM*, 55(4):86–97, 2012.

[Rogers *et al.*, 2011] Alex Rogers, Sasan Maleki, Siddhartha Ghosh, and Jennings Nicholas R. Adaptive Home Heating Control Through Gaussian Process Prediction and Mathematical Programming. In *Proceedings of the Second International Workshop on Agent Technology for Energy Systems (ATES)*, 2011.

[Settles, 2009] B. Settles. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[U. S. Department Of Energy, 2003] U. S. Department Of Energy. Grid 2030: A National Vision For Electricity – Second 100 Years. 2003.

[VDE, 2012] VDE. Demand Side Integration – Lastverschiebungspotenziale in Deutschland. Technical Report, VDE Verband the Elektrotechnik, June 2012.

[Vytelingum *et al.*, 2010] Perukrishnen Vytelingum, Thomas D. Voice, Sarvapali D. Ramchurn, Alex Rogers, and Nicholas R. Jennings. Agent-based Micro-storage Management for the Smart Grid. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2010.

[Weron, 2006] Rafal Weron. *Modeling and Forecasting Electricity Loads and Prices: A Statistical Approach*. HSC Books. Hugo Steinhaus Center, Wroclaw University of Technology, 2006.