Developing a Data-Driven Learning Interest Recommendation System to Promoting Self-Paced Learning on MOOCs

Prepared by Hsuan-Ming Chang

Directed by Prof. Nen-Fu Huang

In Partial Fulfillment of the Requirements

for the Degree of
Master of Science

Department of Computer Science

National Tsing Hua University

Hsinchu, Taiwan 30013, R.O.C.

E-mail: robert-rino@gapp.nthu.edu.tw

May 15, 2016

Abstract

This work proposes a learning-based energy management policy that takes into consideration the trade-off between the depth-of-discharge (DoD) and the lifetime of batteries. The impact of DoD on the energy management policy is often neglected in the past due to the inability to model its effect on the marginal cost per battery usage. In this work, a novel battery cost evaluation method that takes into consideration the DoD of each battery usage is proposed, and is utilized to devise the day-ahead energy management policy using reinforcement learning and linear value-function approximations. The policy determines the amount of energy to purchase for the next day in the day ahead market. A least-square policy iteration (LSPI) with linear approximations of the value function is used to learn the energy management policy. Simulations are provided based on real load profiles, pricing data, and renewable energy arrival statistics. The consideration of the battery cost due to DoD provides a more accurate evaluation of the actual energy cost and leads to an improved energy management policy.

Keywords -Smart grid, energy management system, reinforcement learning, battery, energy storage, depth-of-discharge.

This work was supported in part by the NTHU-Delta Project on "Demand-Side Management in Universities and Enterprise Campuses".

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Massive Online Open Courses (MOOCs) refer to an open educational resources, which allows learners worldwide to take well-designed online courses of interest free of charge. On MOOCs, learners watch the high-quality instructional videos made by professors from prestigious universities, share their ideas and reflections on the discussion forum, and use the online exercise system to evaluate their learning outcome. Due to the fact that the MOOCs provide with high-quality self-directed learning environment without costing much for online learners, MOOCs have been thought of as a contemporary way of 21-century learning.

There are two type of MOOCs, cMOOCs (connectionism MOOCs) and xMOOCs (instructionism MOOCs). These two types of MOOCs are base on different philosophical positions underpinning, cMOOCs focuses on connections between participants in particular on strong content contributions from the participants themselves [1], xMOOCs, by contrast focus on instructor's design of the course. Many famous MOOCs platform such as Coursera[2], edX[3], and Udacity[4] are belong to xMOOCs.

For current xMOOCs, the instructional videos play a significant role in the online learning process [5,6]. In essence, the learning focuses in the form of visual and

audial presented in the instructional videos. Traditionally, video-based learning follows structured instructor-designed sequences for the better results. Owing to the technological nature of the online stream video, it is found that many students drag the play bar replaying specific concept in the video for consolidating their understanding. Therefore, many studies aim to improve the video-based learning environment by adding additional features in video-watching, such as embedded assessment, caption tool, as so on.

In view of the rapid development of data sciences, more and more studies on educational data mining and learning analytics take the advantages of the learners data to optimize learning process. For example, [7] develops a step-by-step annotations feature to improve the learning experience of existing how-to videos. Study [8] constructs a system that recommends students videos best on their forum post, making a self-solved confusion system and meanwhile reducing the teaching load. Therefore, considering the learning needs and the authentic learner data, this study develops a data-driven learning interest recommendation system to promote self-paced learning by integrating educational data mining and word segmentation in the Chinese-speaking MOOC environment. Videomark combines both the learning seek event counts and the subtitles of each video to automatically generate learning concepts for learners in friendly user interface. Through the huge amount of video watching/seeking log data, the Videomark helps learners to quickly identify popular video seek events for consolidating their concept of the learning focus in hope of promoting better self-paced video-based learning environment.

The remainder of this thesis is organized as follows. In Section 2, we first introduce the Markov decision process and brief review the reinforcement learning algorithm which is called least-square policy iteration. In Section 3, the energy

management problem at the consumer side is examined. We designed our system model by considering a EMS center which want to regulate energy flow such as day-ahead energy purchasing, real-time energy purchasing, and energy dumping to minimize marginal cost and prolong battery life. In Section 4, the reinforcement learning based energy management problem is examined. In Section 5, the performance of purposed algorithm is examined.

Chapter 2

Brief Review of Reinforcement Learning and Least-Square Policy Iteration

In this chapter, we present a brief review of the Markov decision process (MDP) and its solution through reinforcement learning. In particular, we consider the least-square policy iteration (LSPI) [?], which utilize linear approximation of value function to approximate the reward of a Markovian transition.

2.1 Markov Decision Process

An MDP is defined by the five tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}', \gamma)$ where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is a finite set of actions, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto [0, 1]$ is the probability model defined such that $\mathcal{P}(s, a, s')$ yields the probability of taking a transition to state s' under action a in state s, $\mathcal{R}' : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}$ is a reward function such that $\mathcal{R}'(s, a, s')$ is the reward of making a transition to state s' when taking action a in state s, and $\gamma \in [0, 1]$ is the discount factor for future reward. We also can define the expected reward for state-action pair (s, a), as $\mathcal{R}(s, a) = \sum_{s' \in \mathcal{S}} \mathcal{P}(s, a, s') \mathcal{R}'(s, a, s')$.

2.2 Least-Square Policy Iteration

After we have the knowledge of MDP, we can introduce the LSPI algorithm. Let $\pi : \mathcal{S} \to \mathcal{A}$ be a deterministic policy that specifies an action $\pi(s)$ for every $s \in \mathcal{S}$. The value function $Q^{\pi}(s, a)$ under the policy π is the expected future reward starting from state s if action a is taken, and can be expressed as

$$Q^{\pi}(s,a) = \mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s,a,s') Q^{\pi}(s', \pi(s')). \tag{2.1}$$

The optimal policy π^* can be found by the policy iteration

$$\pi^{(m+1)}(s) = \arg\max_{a'} Q^{\pi^{(m)}}(s, a') \tag{2.2}$$

for all $s \in \mathcal{S}$, until it converges (i.e., $\pi^* = \pi^{(\infty)}$). However, in most practical cases, the action-value Q^{π} is unknown or is too complicated to compute. In this case, we can adopt a linear approximation, where Q^{π} is approximated by [**?**]

$$\widehat{Q}^{\pi}(s, a; \mathbf{w}) = \sum_{j=1}^{k} \phi_j(s, a)\, w_j \tag{2.3}$$

with $\mathbf{w} = (w_1, w_2, \ldots, w_k)$ being the weight vector and $\{\phi_j\}_{j=1}^{k}$ being an appropriately chosen set of basis functions where k is the number of basis. We require that the basis functions ϕ_j are linearly independent to ensure that there are no redundant parameters and that the matrices involved in the computations are full rank. The way to find the weight is as follow. Let $\mathbf{Q}^{\pi}$ be the column vector of the value function under the policy π with size $|\mathcal{S}||\mathcal{A}|$. Let also $\widehat{\mathbf{Q}}^{\pi}$ be the column vector of the approximated value function that computed by a linear approximation with basis function ϕ_j and parameter w_j. Define $\boldsymbol{\phi}(s, a)$ to be the column vector of basis where each entry j is the corresponding basis function ϕ_j computed at

(s, a). Then, $\widehat{\mathbf{Q}}^\pi$ can be expressed as

$$\widehat{\mathbf{Q}}^\pi = \mathbf{\Phi}\mathbf{w}, \tag{2.4}$$

where $\mathbf{\Phi}$ is a matrix of the form

$$\mathbf{\Phi} = \begin{bmatrix} \boldsymbol{\phi}(s_1, a_1)^T \\ \boldsymbol{\phi}(s_2, a_2)^T \\ \vdots \\ \boldsymbol{\phi}(s_{|\mathcal{S}|}, a_{|\mathcal{A}|})^T \end{bmatrix}. \tag{2.5}$$

To solve the linear approximation, we define the Bellman operator B by

$$B(Q^\pi)(s, a) = \mathcal{R}(s, a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s, a, s') Q^\pi(s', \pi(s')) \tag{2.6}$$

and we can see that the value function $\mathbf{Q}^\pi$ is the fixed point of the Bellman operator, that is,

$$B(\mathbf{Q}^\pi) = \mathbf{Q}^\pi. \tag{2.7}$$

The way to find an approximation is to force the approximate value function to be fixed point under the Bellman operator, that is,

$$B(\widehat{\mathbf{Q}}^\pi) = \widehat{\mathbf{Q}}^\pi. \tag{2.8}$$

For that to be possible, the fixed point has to lie in the space of approximate value functions which is the space spanned by the basis functions. Even though $\widehat{\mathbf{Q}}^\pi$ lies in that space by definition, $B(\widehat{\mathbf{Q}}^\pi)$ may, in general, be out of that space and hence needs to be projected back onto that space. More specifically, the projection finds a least square approximation of $B(\widehat{\mathbf{Q}}^\pi)$ on the space spanned by $\mathbf{\Phi}$, that is, find $\mathbf{w}'$ that minimizes the norm $\|B(\mathbf{\Phi}\mathbf{w}) - \mathbf{\Phi}\mathbf{w}\|$. By the orthogonality principle, we have

$$\mathbf{\Phi}^T(B(\mathbf{\Phi}\mathbf{w}) - \mathbf{\Phi}\mathbf{w}) = 0, \tag{2.9}$$

which leads to

$$\mathbf{w} = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}B(\mathbf{\Phi}\mathbf{w}). \tag{2.10}$$

The fixed-point approximation by using least-square approximation $\mathbf{\Phi}\mathbf{w}'$ is given as

$$\widehat{\mathbf{Q}}^\pi = \mathbf{\Phi}\mathbf{w} \tag{2.11}$$

$$= \mathbf{\Phi}(\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T B(\widehat{\mathbf{Q}}^\pi) \tag{2.12}$$

$$= \mathbf{\Phi}(\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T \left(\mathcal{R} + \gamma\mathcal{P}\widehat{\mathbf{Q}}^\pi\right) \tag{2.13}$$

$$= \mathbf{\Phi}(\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T \left(\mathcal{R} + \gamma\mathcal{P}\mathbf{\Phi}\mathbf{w}\right). \tag{2.14}$$

where $\mathcal{R} + \gamma\mathcal{P}\widehat{\mathbf{Q}}^\pi$ is the matrix form of $B(\widehat{\mathbf{Q}}^\pi)$. Then,

$$\mathbf{\Phi}(\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T \left(\mathcal{R} + \gamma\mathcal{P}\mathbf{\Phi}\mathbf{w}\right) = \mathbf{\Phi}\mathbf{w} \tag{2.15}$$

$$\mathbf{\Phi}\left[(\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T \left(\mathcal{R} + \gamma\mathcal{P}\mathbf{\Phi}\mathbf{w}\right) - \mathbf{w}\right] = 0 \tag{2.16}$$

$$\mathbf{\Phi}^T\left(\mathbf{\Phi} - \gamma\mathbf{P}\mathbf{\Phi}\right)\mathbf{w} = \mathbf{\Phi}\mathcal{R}. \tag{2.17}$$

By solving the above equations, the resultant solution can be expressed as follows,

$$\mathbf{w} = [\mathbf{\Phi}^T(\mathbf{\Phi} - \gamma\mathcal{P}\mathbf{\Phi})]^{-1}\mathbf{\Phi}^T\mathcal{R}. \tag{2.18}$$

Note that the distribution of the approximation error can be control by means of weighted projection. To do this, let ν be a probability distribution over (s, a) and Δ_ν be the diagonal matrix with the projection weights $\nu(s, a)$, then the weighted least-squares fixed point solution is [**?**]

$$\mathbf{w} = [\mathbf{\Phi}^T\Delta_\nu(\mathbf{\Phi} - \gamma\mathcal{P}\mathbf{\Phi})]^{-1}\mathbf{\Phi}^T\Delta_\nu\mathcal{R}, \tag{2.19}$$

where the exact values of $\mathbf{w}$ can be computed by solving the linear system of equations

$$\mathbf{A}\mathbf{w} = \mathbf{c}, \tag{2.20}$$

where $\mathbf{A} = \mathbf{\Phi}^T \Delta_\nu (\mathbf{\Phi} - \gamma \mathcal{P} \mathbf{\Phi})$, $\mathbf{c} = \mathbf{\Phi}^T \Delta_\nu \mathcal{R}$. Since $\mathcal{P}$ and $\mathcal{R}$, in general, is unknown, $\mathbf{A}$ and $\mathbf{c}$ cannot be determined a prior, but they can be learned using samples. The learned linear system can then be solved to yield the learned parameters $\mathbf{w}$. More specifically, we have

$$\mathbf{A} = \mathbf{\Phi}^T \Delta_\nu (\mathbf{\Phi} - \gamma \mathcal{P} \mathbf{\Phi}) \tag{2.21}$$

$$= \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \phi(s,a) \nu(s,a) \left(\phi(s,a) - \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}(s,a,s') \phi(s', \pi(s')) \right)^T \tag{2.22}$$

$$= \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \nu(s,a) \sum_{s' \in \mathcal{S}} \mathcal{P}(s,a,s') [\phi(s,a)(\phi(s,a) - \gamma \phi(s', \pi(s')))^T], \tag{2.23}$$

$$\mathbf{c} = \mathbf{\Phi}^T \Delta_\nu \mathcal{R} \tag{2.24}$$

$$= \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \phi(s,a) \nu(s,a) \sum_{s' \in \mathcal{S}} \mathcal{P}(s,a,s') \mathcal{R}(s,a,s') \tag{2.25}$$

$$= \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \nu(s,a) \sum_{s' \in \mathcal{S}} \mathcal{P}(s,a,s') [\phi(s,a) R(s,a,s')]. \tag{2.26}$$

Since these summations are taken over s, a, and s' and weighted by the projection weight $\nu(s,a)$ and probability $P(s,a,s')$, the special form is that $\mathbf{A}$ is the sum of many rank one matrices of the form

$$\phi(s,a)(\phi(s,a) - \gamma \phi(s', \pi(s')))^T, \tag{2.27}$$

and $\mathbf{c}$ is the sum of many vectors of the form

$$\phi(s,a) R(s,a,s'). \tag{2.28}$$

In the general case, it is impractical to compute this summation over all (s,a,s') triplet. Given any finite set of samples $\mathcal{D} = \{(s_i, a_i, s'_i, r_i)\}_{i=1}^N$ where N is the

number of samples, $\mathbf{A}$ and $\mathbf{c}$ can be learned as

$$\widetilde{\mathbf{A}} = \frac{1}{N}\sum_{i=1}^{N}\left[\phi(s_i,a_i)(\phi(s_i,a_i)-\gamma\phi(s_i',\pi(s_i')))^T\right], \tag{2.29}$$

$$\widetilde{\mathbf{c}} = \frac{1}{N}\sum_{i=1}^{N}\left[\phi(s_i,a_i)r_i\right]. \tag{2.30}$$

By assuming that the distribution $\nu_{\mathcal{D}}$ of the samples in $\mathcal{D}$ over $(\mathcal{S}\times\mathcal{A})$ matches the desired distribution ν. We can rewrite the above equation as

$$\widetilde{\mathbf{A}} = \frac{1}{N}\widetilde{\mathbf{\Phi}}^T(\widetilde{\mathbf{\Phi}}-\gamma\widetilde{\mathcal{P}\Pi_\pi\mathbf{\Phi}}),\ \widetilde{\mathbf{c}} = \frac{1}{N}\widetilde{\mathbf{\Phi}}^T\widetilde{\mathcal{R}} \tag{2.31}$$

where

$$\widetilde{\mathbf{\Phi}} = \begin{pmatrix}\phi(s_1,a_1)^T\\ \vdots \\ \phi(s_N,a_N)^T\end{pmatrix}, \widetilde{\mathcal{P}\Pi_\pi\mathbf{\Phi}} = \begin{pmatrix}\phi(s_1',\pi(s_1'))^T\\ \vdots \\ \phi(s_N',\pi(s_N'))^T\end{pmatrix}, \widetilde{\mathcal{R}} = \begin{pmatrix}r_1\\ \vdots \\ r_N\end{pmatrix}. \tag{2.32}$$

Therefore, as the number of samples tends to infinite, $\widetilde{\mathbf{A}}$ and $\widetilde{\mathbf{c}}$ converge to the matrix of least-square fixed point approximation, which is given as follows

$$\lim_{N\to\infty}\widetilde{\mathbf{A}} = \mathbf{\Phi}^T\Delta_{\nu_{\mathcal{D}}}(\mathbf{\Phi}-\gamma\mathcal{P}\Pi_\pi\mathbf{\Phi}),\ \lim_{N\to\infty}\widetilde{\mathbf{c}} = \mathbf{\Phi}^T\Delta_{\nu_{\mathcal{D}}}\mathcal{R}. \tag{2.33}$$

Let $\widetilde{\mathbf{A}}^{(t)}$ and $\widetilde{\mathbf{c}}^{(t)}$ be the current learned estimates of $\mathbf{A}$ and $\mathbf{c}$ for a policy π, assuming that initially $\widetilde{\mathbf{A}}^{(t)}=0$ and $\widetilde{\mathbf{c}}^{(t)}=0$, A new sample (s_i,a_i,s_i',r_i) contributes to the approximation according to the following update equations is

$$\widetilde{\mathbf{A}}^{(t+1)} = \widetilde{\mathbf{A}}^{(t)} + \phi(s_i,a_i)(\phi(s_i,a_i)-\gamma\phi(s_i',\pi(s_i')))^T, \tag{2.34}$$

$$\widetilde{\mathbf{c}}^{(t+1)} = \widetilde{\mathbf{c}}^{(t)} + \phi(s_i,a_i)r_i. \tag{2.35}$$

We summarize this technique in Algorithm 1.

Algorithm 1: Least Square Policy Iteration (LSPI)

Input: training samples $\mathcal{D} = \{(s_i, a_i, s_i', r_i)\}_{i=1}^{N}$, where N is the number of samples, r_i is the reward of transition (s_i, a_i, s_i'), basis function $\boldsymbol{\phi} = [\phi_1, \ldots, \phi_k]^T$, discount factor γ, and stopping criterion ϵ.

Output: learn the weight vector $\mathbf{w}^*$

$\mathbf{w}^* \leftarrow 0$.
repeat
  $\mathbf{w} \leftarrow \mathbf{w}^*$, $\mathbf{A} \leftarrow 0$, $\mathbf{c} \leftarrow 0$
  for each $(s, a, s', r) \in \mathcal{D}$ **do**
    $\pi(s') \leftarrow \arg\max_{a'} \boldsymbol{\phi}(s', a')^T \mathbf{w}$
    $\mathbf{A} \leftarrow \mathbf{A} + \boldsymbol{\phi}(s, a)(\boldsymbol{\phi}(s, a) - \gamma \boldsymbol{\phi}(s', \pi(s')))^T$
    $\mathbf{c} \leftarrow \mathbf{c} + \boldsymbol{\phi}(s, a) r$
  end
  Solve $\mathbf{A}\mathbf{w}^* = \mathbf{c}$ for $\mathbf{w}^*$.
until $\|\mathbf{w} - \mathbf{w}^*\| < \epsilon$;

Chapter 3

System Model and Problem Formulation

Let us consider a residential home that is connected to the electricity market via the power grid, but is also equipped with renewable energy generators, such as solar panels and wind turbines, and with a battery of size C (kWh) to regulate the energy usage over time. The electricity is purchased from both a day-ahead and a real-time market where the prices may vary over H time slots of the day. The day-ahead and real-time electricity prices are given by $p_d^{\mathrm{DA}}(h)$ and $p_d^{\mathrm{RT}}(h)$ (USD/kWh), respectively, where d is the index of the day and $h \in \{1, 2, \ldots, H\}$ represents the time slot within the day. The user employs an energy management system, as illustrated in Figure 3.1, that determines the amount of day-ahead energy purchase (i.e., the amount of energy to be purchased for the next day) based on the electricity prices, renewable energy arrivals, residential loads and battery levels during the current day. The cost of energy usage consists of the cost of energy purchase from the market as well as the cost of battery usage. First, we formulate the cost of energy purchase from grid and our system model in secton 3.1. Second, in Section 3.2, we introduce number of cycle of batteries and

define the current depth-of-discharge for each time slot, then, formulate the cost of batteries.

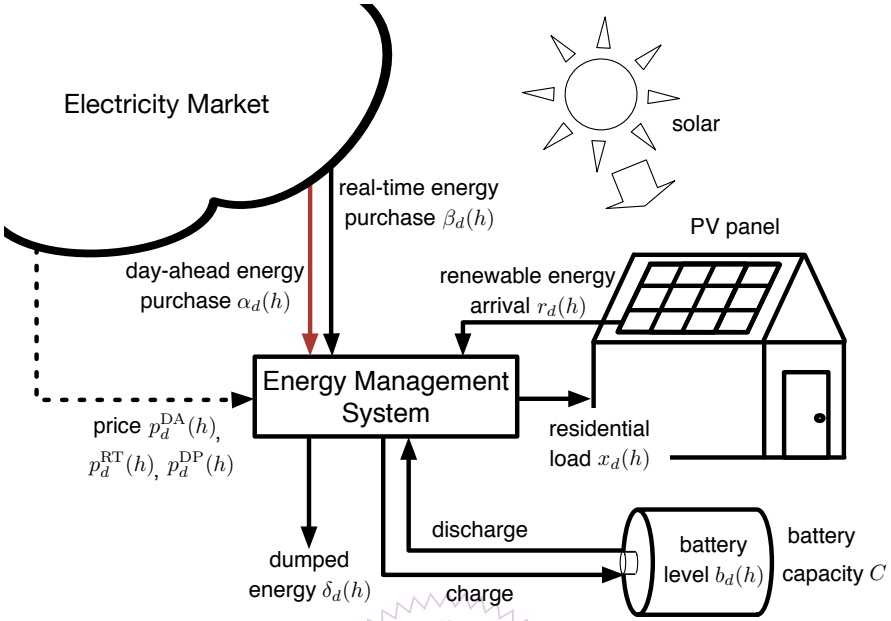

Figure 3.1: System Model.

3.1 Cost of Energy Purchase

The cost of energy purchase from grid can be formulate by the energy flow of EMS. Let $r_d(h)$, $x_d(h)$, and $b_d(h)$ be the amount of renewable energy arrival, residential load, and battery level, respectively, at the end of time slot h on day d, where $h = 1, 2, \ldots, H$ and $d = 1, 2, \ldots$. The amount of day-ahead purchase, denoted by $\alpha_d(h)$, for $h = 1, \ldots, H$, was determined at the end of day $d-1$ based on knowledge of $r_{d-1}(h)$, $x_{d-1}(h)$, and $b_{d-1}(h)$, for all h. Due to the uncertainty of the next-day electricity pricing, renewable energy arrival, and residential loads, the day-ahead purchase may be in excess by amount

$$e_d(h) = b_d(h-1) + r_d(h) + \alpha_d(h) - x_d(h) \tag{3.1}$$

where $b_d(0) \triangleq b_{d-1}(H)$. The intuition of (3.1) is that the possible energy we can utilize, that is, $b_d(h-1)+r_d(h)+\alpha_d(h)$, minus the residential demand $x_d(h)$. Notice that this excess energy can be either positive or negative depending on whether or not energy is over-purchased or under-purchased in the day ahead market. When the excess energy is positive, it is stored in the battery, resulting in battery level of

$$b_d(h) = \max\{0, \min\{e_d(h), C\}\} \tag{3.2}$$

at the end of time slot h. The amount of energy that cannot be stored due to limited battery capacity, i.e.,

$$\delta_d(h) = \max\{0, e_d(h) - C\}, \tag{3.3}$$

is dumped at the price of $p_d^{\mathrm{DP}}(h)$ (USD/kWh). When the excess energy is negative, the amount of energy

$$\beta_d(h) = \max\{0, -e_d(h)\} \tag{3.4}$$

should be purchased from the real-time market to support the current residential demand. The total cost of purchasing electricity from the market is thus given by

$$\kappa_d^{\mathrm{grid}}(h) = p_d^{\mathrm{DA}}(h)\alpha_d(h) + p_d^{\mathrm{RT}}(h)\beta_d(h) + p_d^{\mathrm{DP}}(h)\delta_d(h). \tag{3.5}$$

3.2 Cost of Battery Usage

In addition to the cost of energy purchase from the market, the use of battery to regulate energy purchase and usage over time also results in a cost due to battery degradation. In particular, with each charge and discharge cycle, the battery will degrade progressively and the speed of degradation depends on the DoD [**?**], which is defined as the ratio between the total discharge in a cycle and the total battery

capacity. In fact, the number of cycles that a battery can be used in a lifetime decreases rapidly as the DoD increases and is given by [**?**]

$$N_{\text{cycle}} = c_1 \cdot \text{DoD}^{-c_2} \tag{3.6}$$

for some constants c_1 and c_2 that varies for different type of batteries. For a typical Li-ion battery in 2012, the constants are given by $c_1 = 1331$ and $c_2 = 1.825$ [**?**]. Based on this relation, the residential user may not want to use energy from the battery if the cost of further increasing the DoD is higher than the current electricity price from the market.

To consider the tradeoff between DoD and the battery lifetime, we propose to model the cost of battery usage as the marginal cost of increasing the DoD by the amount of usage. More specifically, let p_{batt} be the battery price per kWh capacity. Then, for a battery with capacity C and a given DoD in each cycle, the average cost per cycle is given by $p^{\text{batt}} \cdot C/(c_1 \cdot \text{DoD}^{-c_2})$. Moreover, by defining the current DoD in time slot h as

$$\text{cDoD}_d(h) = \frac{1}{C} \sum_{t=h'+1}^{h} (b_d(t-1) - b_d(t)) \tag{3.7}$$

where $h' = \max\{1 \le t \le h : b_d(t-1) - b_d(t) < 0\}$[1], the marginal cost of battery usage in time slot h can be modeled as

$$\kappa_d^{\text{batt}}(h) = \max\left\{0, \frac{p^{\text{batt}} \cdot C}{c_1 \cdot \text{cDoD}_d(h)^{-c_2}} - \frac{p^{\text{batt}} \cdot C}{c_1 \cdot \text{cDoD}_d(h-1)^{-c_2}}\right\} \tag{3.8}$$

This novel consideration of the battery cost realistically captures the tradeoff between the DoD and the battery lifetime. The intuition (3.8) is that we consider

[1]For notational simplicity, we assume that the discharge does not extend to the previous day since batteries are usually charged at midnight. However, this can be easily extended to the general case with proper choice of notations.

the cost of battery usage in each time slot by compare there average cost per cycle. Moreover, the cost of battery in (3.8) grow exponentially when current DoD grow linearly.

Hence, the total cost at time h on day d is given by

$$\kappa_d(h) = \kappa_d^{\text{grid}}(h) + \kappa_d^{\text{batt}}(h). \tag{3.9}$$

This cost depends on the day-ahead energy purchase $\boldsymbol{\alpha}_d \triangleq [\alpha_d(1), \ldots, \alpha_d(H)]^T$ as well as the system states, such as electricity prices, renewable energy arrival, residential load profile, and battery level. Different from most works in the literature that rely on (precise or statistical) knowledge of the above-mentioned system states, we utilize a reinforcement learning based policy iteration method to determine the day-ahead energy management policy using only historical data.

Chapter 4

Learning-Based Energy Management Policy with DoD Considerations

In this Section, we specialize the LSPI algorithm to the problem at hand and use it to obtain the desired energy management policy. Specifically, let us define the state of day d as

$$\mathbf{s}_d = (\mathbf{r}_d, \mathbf{x}_d, \mathbf{p}_{d+1}^{\mathrm{DA}}, \mathbf{p}_d^{\mathrm{RT}}, \mathbf{b}_d) \tag{4.1}$$

where $\mathbf{r}_d = [r_d(1), \ldots, r_d(H)]^T$, $\mathbf{x}_d = [x_d(1), \ldots, x_d(H)]^T$, $\mathbf{p}_{d+1}^{\mathrm{DA}} = [p_{d+1}^{\mathrm{DA}}(1), \ldots, p_{d+1}^{\mathrm{DA}}(H)]^T$, $\mathbf{p}_d^{\mathrm{RT}} = [p_d^{\mathrm{RT}}(1), \ldots, p_d^{\mathrm{RT}}(H)]^T$ and $\mathbf{b}_d = [b_d(1), \ldots, b_d(H)]^T$ are the vectors of renewable energy arrivals, residential loads, day-ahead electricity prices, real-time electricity prices, and battery levels, respectively. The action for day d, however, is the decision on the day-ahead purchase for the next day $d+1$, i.e.,

$$\boldsymbol{\alpha}_{d+1} = (\alpha_{d+1}(1), \ldots, \alpha_{d+1}(H))^T. \tag{4.2}$$

The resulting cost is $\kappa_d = \sum_{h=1}^{H} \kappa_d(h)$, where $\kappa_d(h)$ was defined in (3.9), and is taken as minus the reward function mentioned in Chapter 2.

4.1 State Transition

The state transition can be describe as follow. Given state $\mathbf{s}_d = (\mathbf{r}_d, \mathbf{x}_d, \mathbf{p}^{\mathrm{DA}}_{d+1}, \mathbf{p}^{\mathrm{RT}}_d, \mathbf{b}_d)$ and action $\boldsymbol{\alpha}_{d+1}$, the day-ahead purchase for day $d+1$ may be in excess by amount

$$e_{d+1}(h) = b_{d+1}(h-1) + r_{d+1}(h) + \alpha_{d+1}(h) - x_{d+1}(h). \tag{4.3}$$

When the excess energy is positive, it is stored in the battery, resulting in battery level of

$$b_{d+1}(h) = \max\{0, \min\{e_{d+1}(h), C\}\} \tag{4.4}$$

at the end of time slot h. The amount of energy that cannot be stored due to limited battery capacity, i.e.,

$$\delta_{d+1}(h) = \max\{0, e_{d+1}(h) - C\} \tag{4.5}$$

is dumped at the price of $p^{\mathrm{DP}}_{d+1}(h)$ (USD/kWh). When the excess energy is negative, the amount of energy

$$\beta_{d+1}(h) = \max\{0, -e_{d+1}(h)\} \tag{4.6}$$

should be purchased from the real-time market to support the current residential demand. The total cost of purchasing electricity from the market is thus given by

$$\kappa^{\mathrm{grid}}_{d+1}(h) = p^{\mathrm{DA}}_{d+1}(h)\alpha_{d+1}(h) + p^{\mathrm{RT}}_{d+1}(h)\beta_{d+1}(h) + p^{\mathrm{DP}}_{d+1}(h)\delta_{d+1}(h). \tag{4.7}$$

Also, the marginal cost of battery usage is

$$\kappa^{\mathrm{batt}}_{d+1}(h) = \max\left\{0, \frac{p^{\mathrm{batt}} \cdot C}{c_1 \cdot \mathrm{cDoD}_{d+1}(h)^{-c_2}} - \frac{p^{\mathrm{batt}} \cdot C}{c_1 \cdot \mathrm{cDoD}_{d+1}(h-1)^{-c_2}}\right\}. \tag{4.8}$$

The algorithm of evaluating the grid cost and battery cost are summarized in Algorithm 2.

Algorithm 2: Cost Evaluation at Day d

Input: Renewable energy $\mathbf{r}_d$, residential load $\mathbf{x}_d$, electricity day-ahead price $\mathbf{p}_d^{\mathrm{DA}}$, electricity day-ahead price discount $\gamma_{\mathbf{p}^{\mathrm{DA}}}$, electricity real-time price $\mathbf{p}_d^{\mathrm{RT}}$, cost of dumping energy p^{DP}, day-ahead energy purchase $\boldsymbol{\alpha}_d$, battery capacity C (kWh), battery price per kWh p^{batt}, battery parameter c_1 and c_2, remain battery level b_{prev}, last discharge start time $\mathrm{dst}_{\mathrm{prev}}$, and last current DoD $\mathrm{cDoD}_{\mathrm{prev}}$

Output: Excess energy $\mathbf{e}_d$, battery level $\mathbf{b}_d$, real-time energy purchase $\boldsymbol{\beta}_d$, amount of dump energy $\boldsymbol{\delta}_d$, discharge start time dst_d, current DoD cDoD_d, cost form battery $\boldsymbol{\kappa}_d$, cost form grid $\boldsymbol{\kappa}_d$, and total cost κ_d

for h **from** 1 **to** 24 **do**
- $e_d(h) \leftarrow b_{\mathrm{prev}} + r_d(h) + \alpha_d(h) - x_d(h)$
- $b_d(h) \leftarrow \max\{0, \min\{e_d(h), C\}\}$
- $\beta_d(h) \leftarrow \max\{0, -e_d(h)\}$
- $\delta_d(h) \leftarrow \max\{0, e_d(h) - C\}$
- **if** $b_d(h) > b_{prev}$ **then**
  - $\mathrm{ischarge} \leftarrow 1$
- **else**
  - $\mathrm{ischarge} \leftarrow 0$
- **end**
- $\mathrm{dst}_d(h) \leftarrow \mathrm{ischarge} \cdot b_d(h) + \mathrm{ischarge} \cdot \mathrm{dst}_{\mathrm{prev}}$
- $\mathrm{cDoD}_d(h) \leftarrow \frac{\mathrm{dst}_d(h) - b_d(h)}{C}$
- $\kappa_d^{\mathrm{batt}}(h) \leftarrow \max\left\{0, \frac{p^{\mathrm{batt}} \cdot C}{c_1 \cdot \mathrm{cDoD}_d(h)^{-c_2}} - \frac{p^{\mathrm{batt}} \cdot C}{c_1 \cdot \mathrm{cDoD}_{\mathrm{prev}}^{-c_2}}\right\}$
- $\kappa_d^{\mathrm{grid}}(h) \leftarrow \gamma_{\mathbf{p}^{\mathrm{DA}}} \cdot p_d^{\mathrm{DA}}(h) \cdot \alpha_d(h) + p_d^{\mathrm{RT}}(h) \cdot \beta_d(h) + p^{\mathrm{DP}} \cdot \delta_d(h)$
- $\kappa_d(h) \leftarrow \kappa_d^{\mathrm{batt}}(h) + \kappa_d^{\mathrm{grid}}(h)$
- `/* Initial previous battery state */`
- $b_{\mathrm{prev}} \leftarrow b_d(h)$
- $\mathrm{dst}_{\mathrm{prev}} \leftarrow \mathrm{dst}_d(h)$
- $\mathrm{cDoD}_{\mathrm{prev}} \leftarrow \mathrm{cDoD}_d(h)$

end

4.2 Basis

Here, we adopt the LSPI scheme proposed in [**?**] to learn the weight vector $\mathbf{w}$ of the value function approximation in (2.3), that is,

$$\widehat{Q}(\mathbf{s}_d, \boldsymbol{\alpha}_{d+1}; \mathbf{w}^*) = \sum_{j=1}^{k} \phi_j(\mathbf{s}_d, \boldsymbol{\alpha}_{d+1}) w_j. \tag{4.9}$$

The basis is chosen as follows

$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \vdots \\ \phi_{11} \\ \phi_{12} \\ \phi_{13} \\ \phi_{14} \\ \phi_{15} \\ \vdots \\ \phi_{20} \\ \phi_{21} \\ \phi_{22} \\ \phi_{23} \end{bmatrix} = \begin{bmatrix} -\sum_{h=1}^{H} \hat{\kappa}_{d+1}^{\text{grid}}(h) \\ -\sum_{h=1}^{H} \hat{\kappa}_{d+1}^{\text{batt}}(h) \\ \hat{\mathbf{b}}_{d+1}(H) \\ \sum_{h=1}^{4} \mathbf{p}_d^{\text{RT}}(h) \\ \sum_{h=5}^{8} \mathbf{p}_d^{\text{RT}}(h) \\ \sum_{h=9}^{10} \mathbf{p}_d^{\text{RT}}(h) \\ \vdots \\ \sum_{h=19}^{20} \mathbf{p}_d^{\text{RT}}(h) \\ \sum_{h=21}^{24} \mathbf{p}_d^{\text{RT}}(h) \\ \sum_{h=1}^{4} (\mathbf{r}_d(h) - \mathbf{x}_d(h)) \\ \sum_{h=5}^{8} (\mathbf{r}_d(h) - \mathbf{x}_d(h)) \\ \sum_{h=9}^{10} (\mathbf{r}_d(h) - \mathbf{x}_d(h)) \\ \vdots \\ \sum_{h=19}^{20} (\mathbf{r}_d(h) - \mathbf{x}_d(h)) \\ \sum_{h=21}^{24} (\mathbf{r}_d(h) - \mathbf{x}_d(h)) \\ \mathbf{b}_d(H) \\ 1 \end{bmatrix}. \tag{4.10}$$

where $\sum_{h=1}^{H} \hat{\kappa}_{d+1}^{\text{grid}}(h)$, $\sum_{h=1}^{H} \hat{\kappa}_{d+1}^{\text{batt}}(h)$, $\hat{\mathbf{b}}_{d+1}(H)$ are the estimated next-day grid cost, estimated next-day battery cost, and estimated next-day end day battery level, respectively. These estimates can be obtained by the linear prediction of real-life

data and the action $\boldsymbol{\alpha}_{d+1}$, that is,

$$\hat{b}_{d+1}(H) = \max\left\{0, \min\left\{\hat{e}_{d+1}(H), C\right\}\right\}, \tag{4.11}$$

$$\sum_{h=1}^{H} \hat{\kappa}_{d+1}^{\text{grid}}(h) = \sum_{h=1}^{H} \left[p_{d+1}^{\text{DA}}(h)\alpha_{d+1}(h) + \hat{p}_{d+1}^{\text{RT}}(h)\hat{\beta}_{d+1}(h) + p_{d+1}^{\text{DP}}(h)\hat{\delta}_{d+1}(h)\right], \tag{4.12}$$

$$\sum_{h=1}^{H} \hat{\kappa}_{d+1}^{\text{batt}}(h) = \sum_{h=1}^{H} \max\left\{0, \frac{p^{\text{batt}} \cdot C}{c_1 \cdot \widehat{\text{cDoD}}_{d+1}(h)^{-c_2}} - \frac{p^{\text{batt}} \cdot C}{c_1 \cdot \widehat{\text{cDoD}}_{d+1}(h-1)^{-c_2}}\right\}, \tag{4.13}$$

where $\hat{e}_{d+1}(H)$, $\hat{\beta}_d(h)$, $\hat{\delta}_d(h)$, and $\widehat{\text{cDoD}}_d(h)$ are defined by

$$\begin{aligned}
\hat{e}_{d+1}(h) &= \hat{b}_{d+1}(h-1) + \hat{r}_{d+1}(h) + \alpha_{d+1}(h) - \hat{x}_{d+1}(h), h = 1, \ldots, H, \\
\hat{b}_{d+1}(0) &\triangleq b_d(H), \\
\hat{\beta}_d(h) &= \max\{0, -\hat{e}_d(h)\}, \\
\hat{\delta}_d(h) &= \max\{0, \hat{e}_d(h) - C\}, \\
\widehat{\text{cDoD}}_d(h) &= \frac{1}{C}\sum_{t=h'+1}^{h} (\hat{b}_d(t-1) - \hat{b}_d(t)), \\
h' &\triangleq \max\{1 \le t \le h : \hat{b}_d(t-1) - \hat{b}_d(t) < 0\},
\end{aligned}$$

and $\hat{r}_{d+1}(h)$, $\hat{x}_{d+1}(h)$, and $\hat{p}_{d+1}^{\text{RT}}(h)$ can be defined by the linear prediction matrix $\mathbf{F}_d^{\mathbf{r}}$, $\mathbf{F}_d^{\mathbf{x}}$, and $\mathbf{F}_d^{\mathbf{p}^{\text{RT}}}$ (defined in Section 4.3), that is,

$$\hat{\mathbf{r}}_{d+1} = [\hat{r}_{d+1}(1), \ldots, \hat{r}_{d+1}(H)]^T = \mathbf{F}_d^{\mathbf{r}}\mathbf{r}_d, \tag{4.14}$$

$$\hat{\mathbf{x}}_{d+1} = [\hat{x}_{d+1}(1), \ldots, \hat{x}_{d+1}(H)]^T = \mathbf{F}_d^{\mathbf{x}}\mathbf{x}_d, \tag{4.15}$$

$$\hat{\mathbf{p}}_{d+1}^{\text{RT}} = [\hat{\mathbf{p}}_{d+1}^{\text{RT}}(1), \ldots, \hat{\mathbf{p}}_{d+1}^{\text{RT}}(H)]^T = \mathbf{F}_d^{\mathbf{p}^{\text{RT}}}\mathbf{p}_d^{\text{RT}}. \tag{4.16}$$

The choice of these first three basis functions are inspired by the next-state policy iteration (NSPI) proposed in [**?**], where the prediction of the next state was shown to simplify the selection of basis functions. For ϕ_4 to ϕ_{12}, we consider the total

real-time electricity price for current state per 4 hour (from $h = 1$ to $h = 8$ and from $h = 21$ to $h = 24$) and per 2 hour (from $h = 9$ to $h = 20$). Similarly, for ϕ_{13} to ϕ_{21}, we consider the excessed energy for current state per 4 hour (from $h = 1$ to $h = 8$ and from $h = 21$ to $h = 24$) and per 2 hour (from $h = 9$ to $h = 20$). Note that ϕ_{22} and ϕ_{23} are the end day battery level for current state and a constant for linear combination, respectively. Suppose that $\mathbf{w}^*$ is the vector of coefficients obtained after the convergence is reached. Then, the desired energy management policy is obtained as

$$\pi^*(\mathbf{s}_d) = \arg\max_{\boldsymbol{\alpha}'} \widehat{Q}(\mathbf{s}_d, \boldsymbol{\alpha}'; \mathbf{w}^*) = \arg\max_{\boldsymbol{\alpha}'} \sum_{j=1}^{k} \phi_j(\mathbf{s}_d, \boldsymbol{\alpha}') w_j. \tag{4.17}$$

In order to solve $\pi^*(\mathbf{s}_d)$, we adopt a parameter by parameter optimization approach where each entry of $\boldsymbol{\alpha}$ is optimized in turn until no further improvement can be obtained.

4.3 Real-Life Data Prediction

In order to obtain an estimate of the next-day real-life data such as renewable energy, residential load, and real-time electricity price. We do this by taking the linear estimates $\hat{\mathbf{r}}_{d+1} = \mathbf{F}_d^{\mathbf{r}} \mathbf{r}_d$, $\hat{\mathbf{x}}_{d+1} = \mathbf{F}_d^{\mathbf{x}} \mathbf{x}_d$, and $\hat{\mathbf{p}}_{d+1}^{\mathrm{RT}} = \mathbf{F}_d^{\mathbf{p}^{\mathrm{RT}}} \mathbf{p}_d^{\mathrm{RT}}$, where $\hat{\mathbf{r}}_{d+1}$, $\hat{\mathbf{x}}_{d+1}$, and $\hat{\mathbf{p}}_{d+1}^{\mathrm{RT}}$ are the predicted next-day renewable arrival, residential load, and real-time electricity price, respectively. Then, the linear estimator for day $d + 1$,

$\mathbf{F}_d^{\mathbf{r}}$, $\mathbf{F}_d^{\mathbf{x}}$, and $\mathbf{F}_d^{\mathbf{p}^{\mathrm{RT}}}$, can be found by taking the least square solution, i.e.,

$$\mathbf{F}_d^{\mathbf{r}*} = \arg\min_{\mathbf{F}} \sum_{i=1}^{d-1} \|\mathbf{r}_{i+1} - \mathbf{F}\mathbf{r}_i\|^2, \tag{4.18}$$

$$\mathbf{F}_d^{\mathbf{x}*} = \arg\min_{\mathbf{F}} \sum_{i=1}^{d-1} \|\mathbf{x}_{i+1} - \mathbf{F}\mathbf{x}_i\|^2, \tag{4.19}$$

$$\mathbf{F}_d^{\mathbf{p}^{\mathrm{RT}}*} = \arg\min_{\mathbf{F}} \sum_{i=1}^{d-1} \|\mathbf{p}_{i+1}^{\mathrm{RT}} - \mathbf{F}\mathbf{p}_i^{\mathrm{RT}}\|^2. \tag{4.20}$$

To solve the least square solution, for $\mathbf{F}_d^{\mathbf{r}*}$, we take the derivate, i.e.,

$$\frac{\partial}{\partial \mathbf{F}} \sum_{i=1}^{d-1} \|\mathbf{r}_{i+1} - \mathbf{F}\mathbf{r}_i\|^2 = \frac{\partial}{\partial \mathbf{F}} \sum_{i=1}^{d-1} [\mathbf{r}_{i+1}^T\mathbf{r}_{i+1} - \mathbf{r}_{i+1}^T\mathbf{F}\mathbf{r}_i - \mathbf{r}_i^T\mathbf{F}^T\mathbf{r}_{i+1} + \mathbf{r}_i^T\mathbf{F}^T\mathbf{F}\mathbf{r}_i] = 0 \tag{4.21}$$

which imply

$$\sum_{i=1}^{d-1} [-\mathbf{r}_{i+1}\mathbf{r}_i^T - \mathbf{r}_{i+1}\mathbf{r}_i^T + \mathbf{F}(\mathbf{r}_i\mathbf{r}_i^T + \mathbf{r}_i\mathbf{r}_i^T)] = 0, \tag{4.22}$$

and

$$\mathbf{F}_d^{\mathbf{r}*} = \left[\sum_{i=1}^{d-1} \mathbf{r}_{i+1}\mathbf{r}_i^T\right] \left[\sum_{i=1}^{d-1} \mathbf{r}_i\mathbf{r}_i^T\right]^{-1}. \tag{4.23}$$

Similarly, for $\mathbf{F}^{\mathbf{x}}$ and $\mathbf{F}^{\mathbf{p}^{\mathrm{RT}}}$, we have

$$\mathbf{F}_d^{\mathbf{x}*} = \left[\sum_{i=1}^{d-1} \mathbf{x}_{i+1}\mathbf{x}_i^T\right] \left[\sum_{i=1}^{d-1} \mathbf{x}_i\mathbf{x}_i^T\right]^{-1}, \tag{4.24}$$

and

$$\mathbf{F}_d^{\mathbf{p}^{\mathrm{RT}}*} = \left[\sum_{i=1}^{d-1} \mathbf{p}_{i+1}^{\mathrm{RT}}\mathbf{p}_i^{\mathrm{RT}T}\right] \left[\sum_{i=1}^{d-1} \mathbf{p}_i^{\mathrm{RT}}\mathbf{p}_i^{\mathrm{RT}T}\right]^{-1}. \tag{4.25}$$

Moreover, due to non-stationary of mean of data, we need to pre-process the data before we evaluate estimator. First, assuming that the data is stationary in the past 30 day, then we remove the mean of the past 30 day for each day, i.e., let

$$\tilde{\mathbf{r}}_d \triangleq \mathbf{r}_d - \frac{1}{30} \sum_{j=d-(30-1)}^{d} \mathbf{r}_j. \tag{4.26}$$

Second, we only consider the data whose daily total value without mean is in the 99.7% confidence interval in the past 30 day, i.e.,

$$\tilde{\mathbf{r}}'_d \triangleq \sum_{h=1}^{H} \tilde{\mathbf{r}}_d(h) \tag{4.27}$$

and

$$\tilde{\mu}'_d \triangleq \frac{1}{30} \sum_{j=d-(30-1)}^{d} \tilde{\mathbf{r}}'_j, \tag{4.28}$$

$$\tilde{\sigma}'_d \triangleq \sqrt{\frac{1}{30} \sum_{j=d-(30-1)}^{d} (\tilde{\mathbf{r}}'_j - \tilde{\mu}'_j)^2}. \tag{4.29}$$

Then, the training data will exclude the data at day d if $\tilde{\mathbf{r}}'_d > \tilde{\mu}'_d + 3\tilde{\sigma}'_d$. Therefore, for some day $d+1$, the estimator for renewable energy is

$$\mathbf{F}_d^{\mathbf{r}*} = \left[\sum_{i=1}^{d-1} \tilde{\mathbf{r}}_{i+1} \tilde{\mathbf{r}}_i^T\right] \left[\sum_{i=1}^{d-1} \tilde{\mathbf{r}}_i \tilde{\mathbf{r}}_i^T\right]^{-1} \tag{4.30}$$

where the summation in (4.30) ignore the index i if $\tilde{\mathbf{r}}'_i > \tilde{\mu}'_i + 3\tilde{\sigma}'_i$ or $\tilde{\mathbf{r}}'_{i+1} > \tilde{\mu}'_{i+1} + 3\tilde{\sigma}'_{i+1}$ since they are not a good linear transition. Then, for some day $d+1$, the predicted renewable energy is

$$\hat{\mathbf{r}}_{d+1} = \mathbf{F}_d^{\mathbf{r}*} \tilde{\mathbf{r}}_d + \frac{1}{30} \sum_{j=d-(30-1)}^{d} \mathbf{r}_j. \tag{4.31}$$

Similarly, for residential load and real-time electricity price, we have

$$\hat{\mathbf{x}}_{d+1} = \mathbf{F}_d^{\mathbf{x}*} \tilde{\mathbf{x}}_d + \frac{1}{30} \sum_{j=d-(30-1)}^{d} \mathbf{x}_j, \tag{4.32}$$

$$\hat{\mathbf{p}}_{d+1}^{\mathrm{RT}} = \mathbf{F}_d^{\mathbf{p}^{\mathrm{RT}}*} \tilde{\mathbf{p}}_d^{\mathrm{RT}} + \frac{1}{30} \sum_{j=d-(30-1)}^{d} \mathbf{p}_j^{\mathrm{RT}}. \tag{4.33}$$

where

$$\mathbf{F}_d^{\mathbf{x}*} = \left[\sum_{i=1}^{d-1} \tilde{\mathbf{x}}_{i+1}\tilde{\mathbf{x}}_i^T\right]\left[\sum_{i=1}^{d-1} \tilde{\mathbf{x}}_i\tilde{\mathbf{x}}_i^T\right]^{-1}, \tag{4.34}$$

$$\mathbf{F}_d^{\mathbf{p}^{\mathrm{RT}}*} = \left[\sum_{i=1}^{d-1} \tilde{\mathbf{p}}_{i+1}^{\mathrm{RT}}(\tilde{\mathbf{p}}_i^{\mathrm{RT}})^T\right]\left[\sum_{i=1}^{d-1} \tilde{\mathbf{p}}_i^{\mathrm{RT}}(\tilde{\mathbf{p}}_i^{\mathrm{RT}})^T\right]^{-1}. \tag{4.35}$$

and

$$\tilde{\mathbf{x}}_i \triangleq \mathbf{x}_i - \frac{1}{30}\sum_{j=i-(30-1)}^{i} \mathbf{x}_j, \tag{4.36}$$

$$\tilde{\mathbf{p}}_i^{\mathrm{RT}} \triangleq \mathbf{p}_i^{\mathrm{RT}} - \frac{1}{30}\sum_{j=i-(30-1)}^{i} \mathbf{p}_j^{\mathrm{RT}}. \tag{4.37}$$

Specifically, since we have perfect knowledge of day-ahead electricity price for next day. We can combine the information from day-ahead price. Let $w_{\mathbf{p}}$ be the weight of information from real-time price, then $1-w_{\mathbf{p}}$ is the weight of information from day-ahead price. So, the real-time prediction method, so called weighted average method, can be rewritten by

$$\hat{\mathbf{p}}_{d+1}^{\mathrm{RT}} = w_{\mathbf{p}}\left(\mathbf{F}_d^{\mathbf{p}^{\mathrm{RT}}*}\tilde{\mathbf{p}}_d^{\mathrm{RT}} + \frac{1}{30}\sum_{j=d-(30-1)}^{d} \mathbf{p}_j^{\mathrm{RT}}\right) + (1-w_{\mathbf{p}})\mathbf{p}_{d+1}^{\mathrm{DA}}. \tag{4.38}$$

where $\mathbf{p}_{d+1}^{\mathrm{DA}}$ is the day-ahead electricity price at day $d+1$. We can see the performance comparison in Figure. **??**.

The prediction error is computed by normalize mean square error (NMSE) for each year, i.e.,

$$\mathrm{NMSE} = \frac{\frac{1}{365}\sum_{d=1}^{365}\frac{1}{24}\sum_{h=1}^{24}(\hat{x}_d(h)-x_d(h))^2}{\frac{1}{365}\sum_{d=1}^{365}\frac{1}{24}\sum_{h=1}^{24}(\bar{x}(h)-x_d(h))^2} = \frac{\sum_{d=1}^{365}\sum_{h=1}^{24}(\hat{x}_d(h)-x_d(h))^2}{\sum_{d=1}^{365}\sum_{h=1}^{24}(\bar{x}(h)-x_d(h))^2}.$$

The NMSE is fair to compare the estimator for different data since variance are much different in different data or years. To see the long-term performance, we

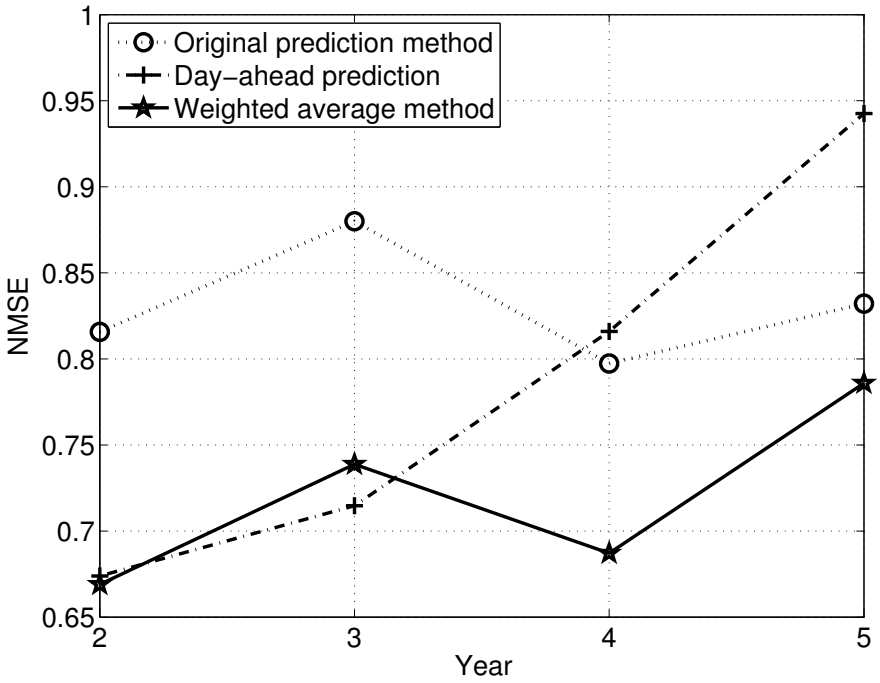

Figure 4.1: Prediction Performance (Electricity Price).

generate also 5 years of synthesized data obtained with random linear combinations of the data from 2010 to 2014, i.e.,

$$\mathbf{x}_{365\times(y-1)+d} = \sum_{i=1}^{5} z_i \mathbf{x}_{365\times(i-1)+d} \tag{4.39}$$

where $d = 1, \ldots, 365$, $y = 6, \ldots, 10$, and z_i are random generated from uniform distribution in $[0, 1]$ and normalize such that $z_1 + \ldots + z_5 = 1$. The detail method is shown in algorithm **??**.

In the end of this chapter, we summarize the proposed algorithm in Algorithm 4. In step 1, we initial the policy for different year. The first year is the training data which is applied equal cost policy and 6 random policy (equal cost policy and random policy will introduce in Chapter 5). The year e is used to enhance the training data in first year since we utilize the weight obtained by first year to testing the training data. After the second year, we apply LSPI with

Algorithm 3: Synthesized Data

1 **for each** $y \in \{6, 7, 8, 9, 10\}$ **do**
2 $\quad z_i = \mathcal{U}(0, 1), i = 1, \ldots, 5$
3 $\quad z_i = z_i / \sum_{i'=1}^{5} z'_i, i = 1, \ldots, 5$
4 $\quad$ **for each** $d \in \{1, 2, \ldots, 365\}$ **do**
5 $\quad\quad \mathbf{x}_{365\times(y-1)+d} = \sum_{i=1}^{5} z_i \mathbf{x}_{365\times(i-1)+d}$
6 $\quad$ **end**
7 **end**

perfect prediction in our training data. In step 2, we initial the previous state for different year since the first states are not the same in different year (the first state in the second year is obtained by equal cost policy and the first state in the third year is obtained by LSPI). In step 3, we generate the samples in each year and testing except first year. This step is a simple loop to generate the triple $(\mathbf{s}_d, \boldsymbol{\alpha}_{d+1}, \mathbf{s}_{d+1})$ by using different policy. In step 5, we adopt the LSPI algorithm to obtain the weight of basis.

Algorithm 4: EMS-LSPI

for each $y \in \{1, e, 2, 3, 4, 5\}$ **do**
  /* Step 1: Initial policy name */
  if $y = 1$ **then**
    $\Pi \leftarrow$ {equal cost policy, 6 random policy}
  else if $y = e$ **then**
    $\Pi \leftarrow$ {LSPI, 6 random policy}
  else
    $\Pi \leftarrow$ {LSPI, LSPI with perfect prediction, equal cost policy, 6 random policy}
  end
  /* Step 2: Initial previous state */
  if $y = 1$ **or** $y = e$ **then**
    $\mathbf{s}_{\text{prev}} \leftarrow (\mathbf{r}_1, \mathbf{x}_1, \mathbf{p}_2^{\text{DA}}, \mathbf{p}_1^{\text{RT}}, \mathbf{b}_1)$ where $b_1(h) = C$ for all h.
  else if $y = 2$ **then**
    $\mathbf{s}_{\text{prev}} \leftarrow (\mathbf{r}_d, \mathbf{x}_d, \mathbf{p}_{d+1}^{\text{DA}}, \mathbf{p}_d^{\text{RT}}, \mathbf{b}_d^{\pi})$ where $d = 365 \times (y-1) + 1$ and π is equal cost policy.
  else
    $\mathbf{s}_{\text{prev}} \leftarrow (\mathbf{r}_d, \mathbf{x}_d, \mathbf{p}_{d+1}^{\text{DA}}, \mathbf{p}_d^{\text{RT}}, \mathbf{b}_d^{\pi})$ where $d = 365 \times (y-1) + 1$ and π is LSPI.
  end
  /* Step 3: Generate samples (each year) and testing (except first year) */
  for each $d \in \{365 \times (y-1) + 1, \ldots, 365 \times (y-1) + 365\}$ **do**
    /* Set state for day d (current state) */
    $\mathbf{s}_d \leftarrow \mathbf{s}_{\text{prev}}$
    /* Set action for next day */
    for each $\pi \in \Pi$ **do**
      if $\pi =$ *LSPI* **then**
        $\boldsymbol{\alpha}_{d+1}^{\pi} \leftarrow \arg\max_{\boldsymbol{\alpha}'} \widehat{Q}(T_r(\mathbf{s}_d, \boldsymbol{\alpha}'), T_x(\mathbf{s}_d, \boldsymbol{\alpha}'), \mathbf{p}_{d+1}^{\text{DA}}, T_{p^{\text{RT}}}(\mathbf{s}_d, \boldsymbol{\alpha}'), T_b(\mathbf{s}_d, \boldsymbol{\alpha}'), \boldsymbol{\alpha}_{d+1}, \mathbf{s}_d; \mathbf{w}^*)$
      else if $\pi =$ *LSPI with perfect prediction* **then**
        $\boldsymbol{\alpha}_{d+1}^{\pi} \leftarrow \arg\max_{\boldsymbol{\alpha}'} \widehat{Q}(T_r(\mathbf{s}_d, \boldsymbol{\alpha}'), T_x(\mathbf{s}_d, \boldsymbol{\alpha}'), \mathbf{p}_{d+1}^{\text{DA}}, T_{p^{\text{RT}}}(\mathbf{s}_d, \boldsymbol{\alpha}'), T_b(\mathbf{s}_d, \boldsymbol{\alpha}'), \boldsymbol{\alpha}_{d+1}, \mathbf{s}_d; \mathbf{w}^*)$
      else if $\pi =$ *equal cost policy* **then**
        $\boldsymbol{\alpha}_{d+1}^{\pi}(h) \leftarrow \frac{1/\hat{p}_{d+1}^{\text{DA}}(h)}{\sum_{h_2=1}^{24} 1/\hat{p}_{d+1}^{\text{DA}}(h_2)} \sum_{h_1=1}^{24} (\hat{x}_{d+1}(h_1) - \hat{r}_{d+1}(h_1))$
      else if $\pi =$ *random policy* **then**
        $\boldsymbol{\alpha}_{d+1}^{\pi}(h) \leftarrow \mathcal{U}(\mu_{d+1}^{\text{LSPI}}(h) - 1.5\sigma_{d+1}^{\text{LSPI}}(h), \mu_{d+1}^{\text{LSPI}}(h) + 1.5\sigma_{d+1}^{\text{LSPI}}(h))$ where π' is equal cost policy if $y = 1$, otherwise, π' is the LSPI policy.
      end
      $\mathbf{s}'^{\pi}_{d+1} \leftarrow (\mathbf{r}_{d+1}, \mathbf{x}_{d+1}, \mathbf{p}_{d+2}^{\text{DA}}, \mathbf{p}_{d+1}^{\text{RT}}, \mathbf{b}_{d+1}^{\pi})$; /* Set state for next day (next state) */
      $r_{d+1}^{\pi} \leftarrow -\kappa_{d+1}$; /* Set reward for next day (for this transition) */
    end
    /* Initial previous state */
    if $y = 1$ **then**
      $\mathbf{s}_{\text{prev}} \leftarrow \mathbf{s}'^{\pi}_{d+1}$ where π is equal cost policy.
    else
      $\mathbf{s}_{\text{prev}} \leftarrow \mathbf{s}'^{\pi}_{d+1}$ where π is LSPI.
    end
  end
  /* Step 4: LSPI algorithm */
  if $y = 5$ **then**
    break for loop ; /* No need to training last year */
  end
  $\mathcal{D} \leftarrow \{(\mathbf{s}_d, \boldsymbol{\alpha}_{d+1}^{\pi}, \mathbf{s}'^{\pi}_{d+1}, r_{d+1}^{\pi})\}_{\pi \in \Pi, d \in \{1,2,\ldots,365\times(y-1)+365\}}$
  $\mathbf{w}^* \leftarrow \text{LSPI}(\mathcal{D}, \boldsymbol{\phi}, \gamma, \epsilon)$
end

Chapter 5

Simulation

In this section, simulations are performed using real-life data to evaluate the performance of the proposed energy management policy. In the experiments, the hourly residential load $x_d(h)$, for $h = 1, \ldots, H$, is based on the average hourly load data of a class of residential single families with electric space heat delivery (The definition can be found in ComEd's electronic tariff documents [1]) during January 1, 2010 to December 31, 2014, provided by ComEd [**?**]. In Figure 5.1, we can see that the residential load in the winter (January, February, and December) in 2010 is higher then other season in 2010 because of the electric space heat delivery. The day-ahead electricity price is based on the data from January 1, 2010 to December 31, 2014 provided by the ComEd RRTP program [**?**]. In Figure 5.2, we can see that the real-time electricity price in summer (January, February, and December) is more expensive then other season.

[1] Residential Single Family With Electric Space Heat Delivery Class means the delivery class applicable to any retail customer in the residential sector (a) that uses electric service for residential purposes, (b) for which service is provided through a separate meter from an overhead or underground connection that serves no more than two (2) retail customers, and (c) that uses only (i) electric resistance heating devices, (ii) electric-only heat pumps, (iii) solar energy collectors that provide space heating through heat exchangers, or (iv) any combination of the preceding items (i) through (iii) to meet the entire space heating requirements at such retail customer's premises.

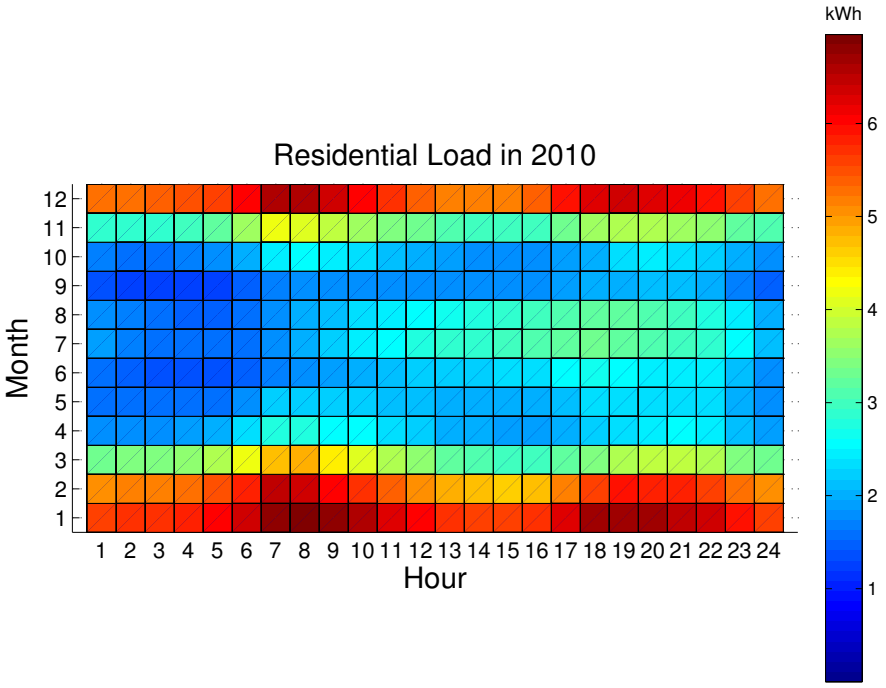

Figure 5.1: Average residential load per month in 2010.

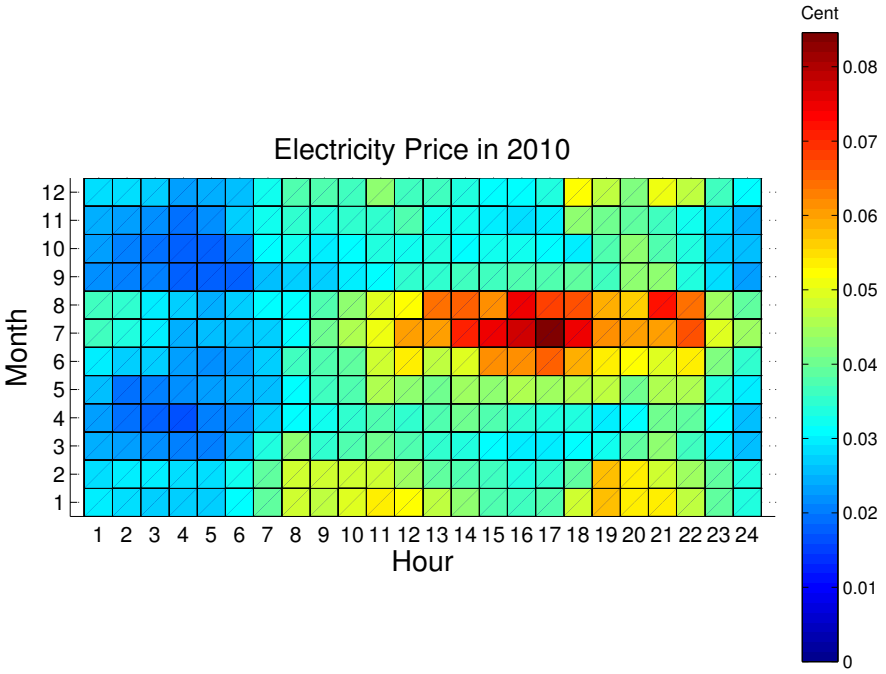

Figure 5.2: Average electricity price per month in 2010.

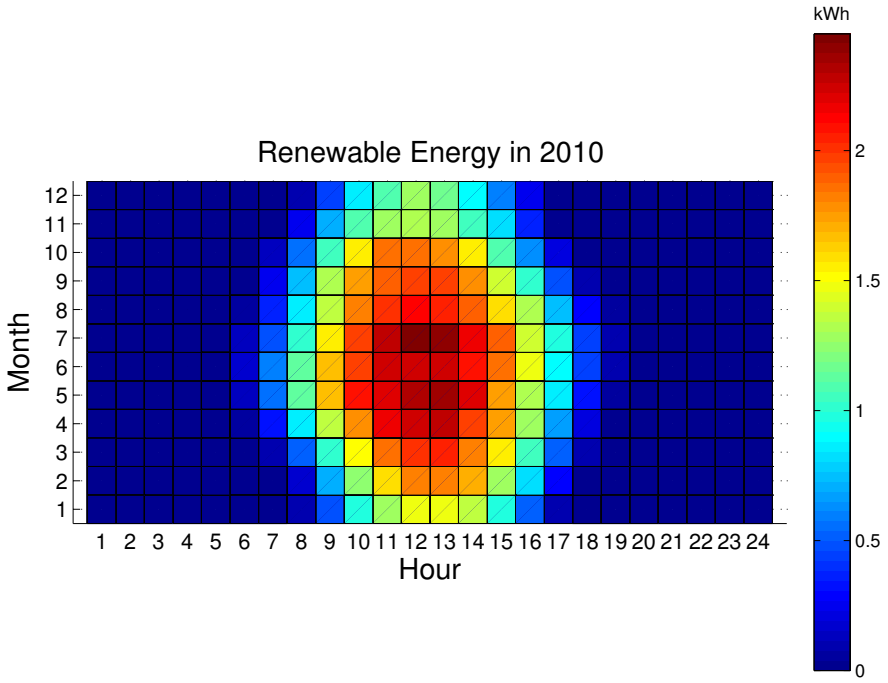

Figure 5.3: Average renewable energy arrival (solar panel) per month in 2010.

Note that the concept of day-ahead purchase has not yet been adopted in practice in the residential setting. The day-ahead electricity price given by [**?**] is actually a prediction of real-time electricity price that the company gives the consumer for reference. In our simulations, we take 80% of the day-ahead electricity price given by [**?**] as the price for day-ahead purchase in our problem. Therefore, the grid cost is given by

$$\kappa_d^{\text{grid}}(h) = 0.8 p_d^{\text{DA}}(h)\alpha_d(h) + p_d^{\text{RT}}(h)\beta_d(h) + p_d^{\text{DP}}(h)\delta_d(h). \tag{5.1}$$

Moreover, we assume that renewable energy is obtained from photo-voltaic (PV) solar panels and the energy arrival is computed using the NREL PVWatts Calculator [**?**] with Chicago's public weather data (TMY2, TMY3). The size of the PV panel is 25 m^2. Here, we set $H = 24$ and $p_d^{\text{DP}}(h) = 0.5$ Cent per kWh. In Figure 5.3, we can see that the hours of sunshine in summer is more than winter.

In the LSPI, the data from the first year, i.e., 2010, along with actions obtained from equal cost policy and 6 random policy are used as the training data samples. In the equal cost policy, the action at time h of day $d+1$ is chosen as

$$\alpha_{d+1}(h) = \frac{1/\hat{p}^{\mathrm{DA}}_{d+1}(h)}{\sum_{h_2=1}^{24} 1/\hat{p}^{\mathrm{DA}}_{d+1}(h_2)} \sum_{h_1=1}^{24} (\hat{x}_{d+1}(h_1) - \hat{r}_{d+1}(h_1)) \tag{5.2}$$

so that the anticipated cost is equal in each time slot. In the random policy, the action at time h of day $d+1$ is chosen as

$$\alpha_{d+1}(h) \sim \mathcal{U}(\mu^{\mathrm{LSPI}}_{d+1}(h) - 1.5\sigma^{\mathrm{LSPI}}_{d+1}(h), \mu^{\mathrm{LSPI}}_{d+1}(h) + 1.5\sigma^{\mathrm{LSPI}}_{d+1}(h)) \tag{5.3}$$

where $\mu^{\mathrm{LSPI}}_{d+1}$ is the mean of action of all LSPI policy before day $d+1$ at time slot h and $\sigma^{\mathrm{LSPI}}_{d+1}$ is the stander deviation of action of all LSPI policy before day $d+1$ at time slot h. The discount factor is $\gamma = 0.9$. The proposed algorithm is summarized in Algorithm 4.

In Figure 5.4, the average cost per year is shown for 5 different policies, namely, (i) equal cost, (ii) LSPI without DoD considerations, (iii) LSPI with DoD considerations (i.e., the proposed scheme), (iv) perfect prediction for LSPI without DoD consideration, and (v) perfect prediction for LSPI with DoD consideration policies. In (ii) and (iv), the policy is determined by using LSPI independently of the battery cost; in (iv) and (v), the policy is determined with perfect prediction of the next state and, thus, serves as a lower bound to the achievable performance. We can see the proposed LSPI with DoD considerations provides significant reduction in energy cost per year compared to policies (i) and (ii) that do not take into consideration the battery cost due to DoD. However, when there is prediction error, the policies become more conservative in their actions, even in the without DoD case. Therefore, when there is prediction error, the battery cost for the without

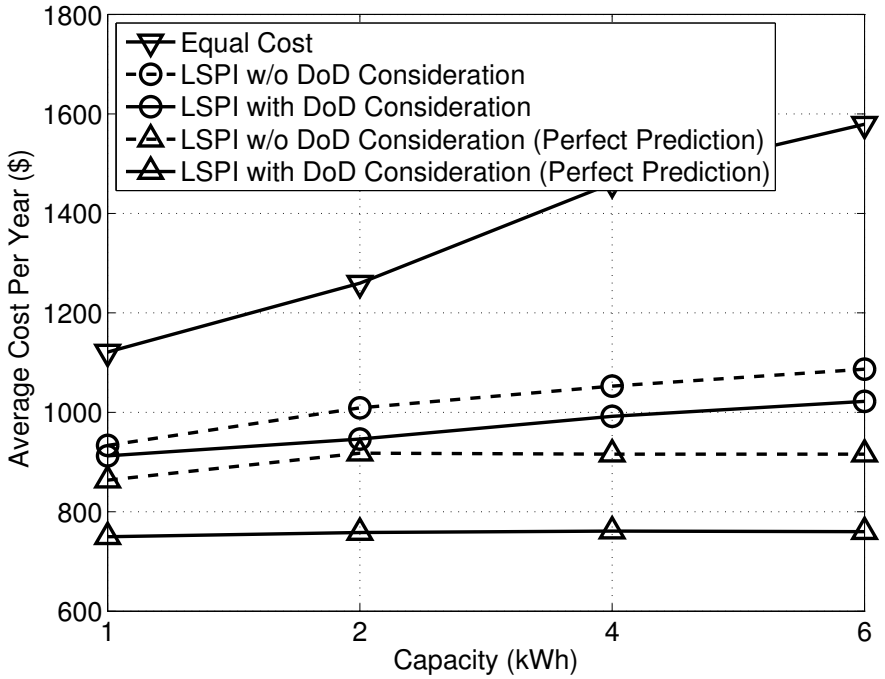

Figure 5.4: Average cost per year versus different battery sizes.

DoD case does not increase so drastically as the battery size increases. In Figure 5.5, we can see that we have significant performance improve in battery cost while LSPI takes into consideration the battery cost due to DoD. In Figure 5.6, we expect that the police use more less grid since it is independently of the battery cost in without DoD case. Being conservative helps prevent deep discharge in the without DoD case when there is prediction error. However, in the without DoD case, it may still discharge more frequently than the with DoD case. In the case with prediction errors, the without DoD case may discharge prematurely, causing its grid cost to also be higher. However, in the with DoD case, the battery does not discharge as deeply as in the without DoD case and, thus, saves energy in the battery for use when the price is unexpectedly high.

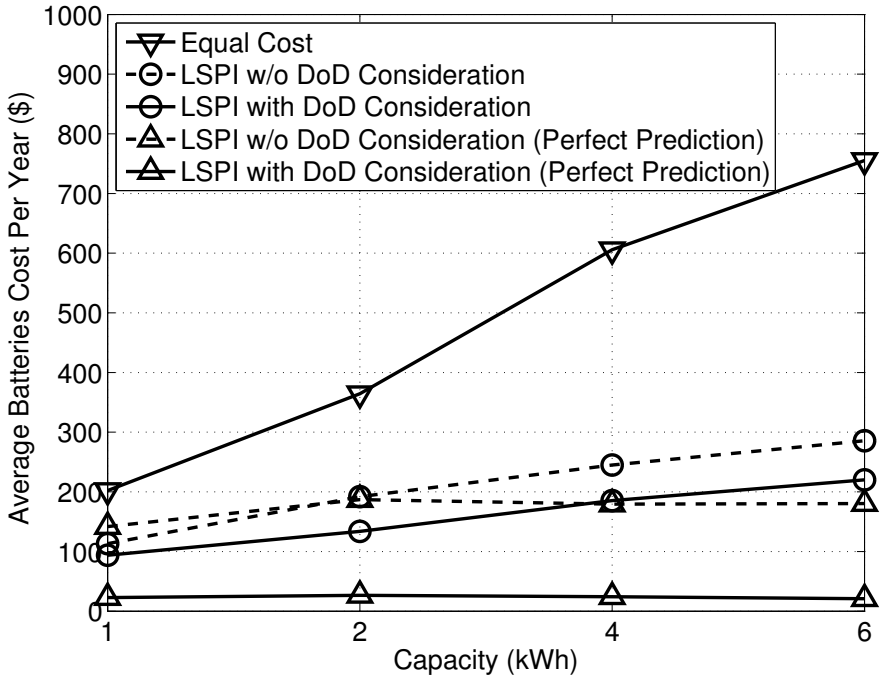

Figure 5.5: Average battery cost per year versus different battery sizes.

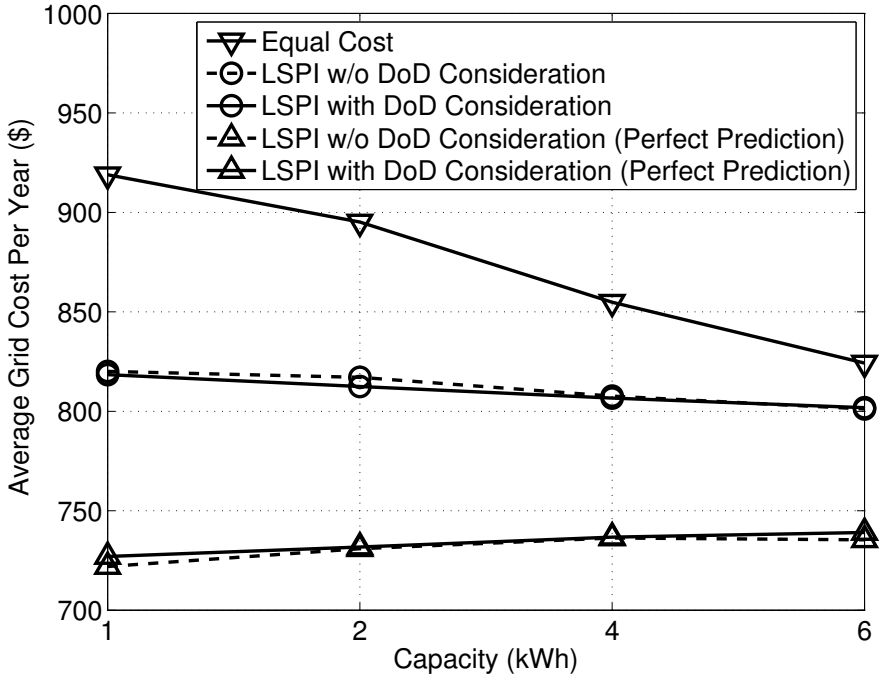

Figure 5.6: Average grid cost per year versus different battery sizes.

In Figures 5.7 and 5.8, we shown the total cost with different battery size in 5 years. We can see that total cost in 5 different policies are increase rapidly from 3-th years to 5-th years because of the variation electricity price in 5-th year is highest then others and variation electricity price in 3-th year is lowest then others.

In Figure 5.9, we show the number of battery replacements that are needed in a 10 year period versus the battery parameter c_2. The 10 year period includes real data from 2011 to 2014 (with 2010 and 2011 as training data) and also 5 years of synthesized data obtained with random linear combinations of the data from 2010 to 2014. The battery size is fixed as $C = 4$ kWh. We can see that the number of battery replacement increases as the parameter c_2 decreases since the number of cycles decreases faster with DoD in this case. Moreover, the proposed LSPI with DoD considerations successfully prolongs the lifetime of batteries compared to other schemes.

In Figure 5.10, we show the average cost per year versus the battery parameter c_2. The battery size is fixed as $C = 4$ kWh. We can see that the average cost increase as the parameter c_2 decrease since the battery cost is increase faster with DoD in this case. This result is with respect to the increase of the number of battery replacement since since the number of cycles decreases faster with DoD. In Figure 5.11, the parameter is changed to p^{batt}. We can see that the average cost per year is increase when battery price increase since the cost of each battery replacement is increase.

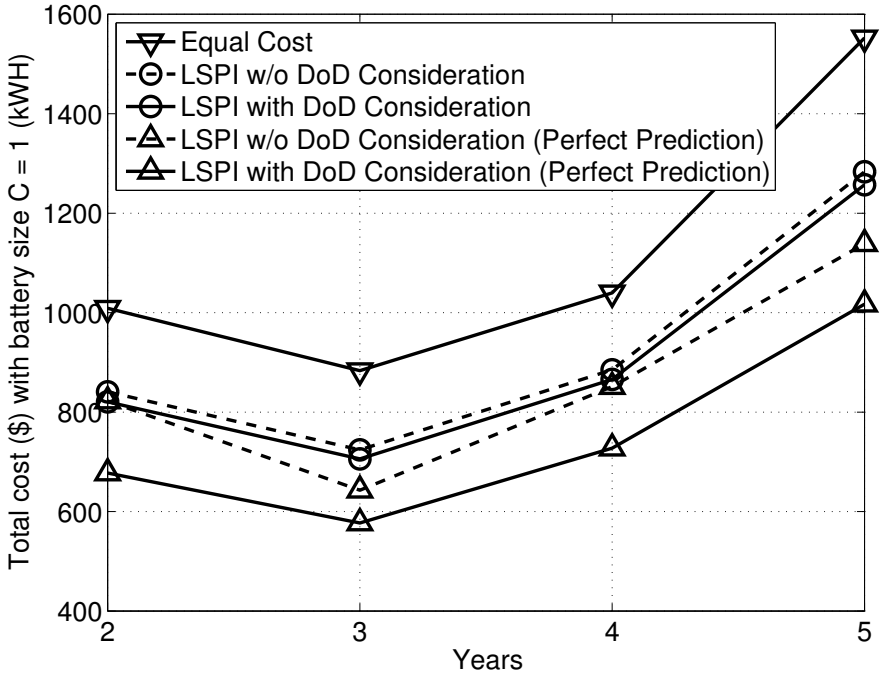

Figure 5.7: Total cost with battery size $C = 1$ (kWH) versus different years.

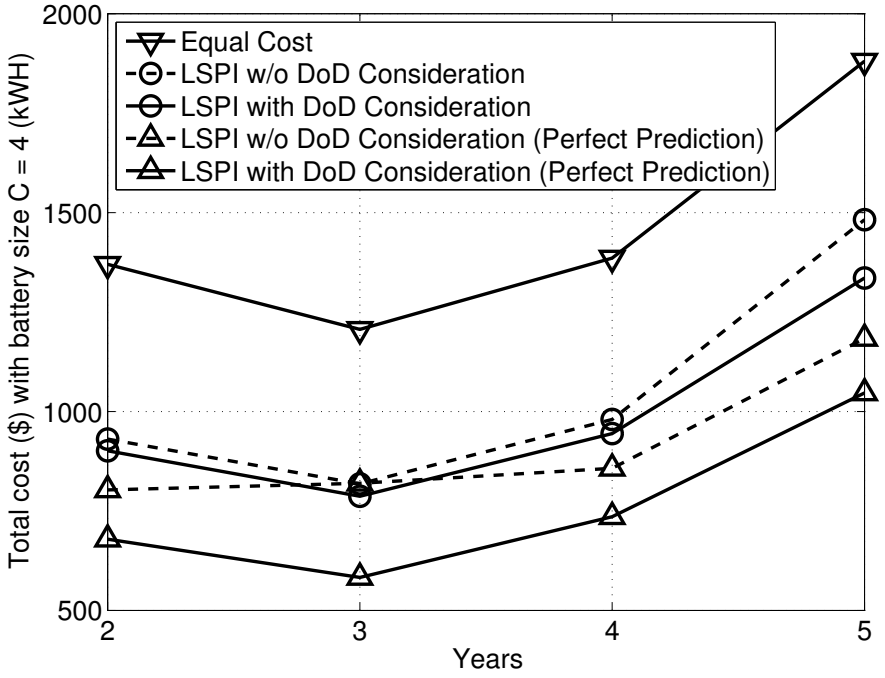

Figure 5.8: Total cost with battery size $C = 4$ (kWH) versus different years.

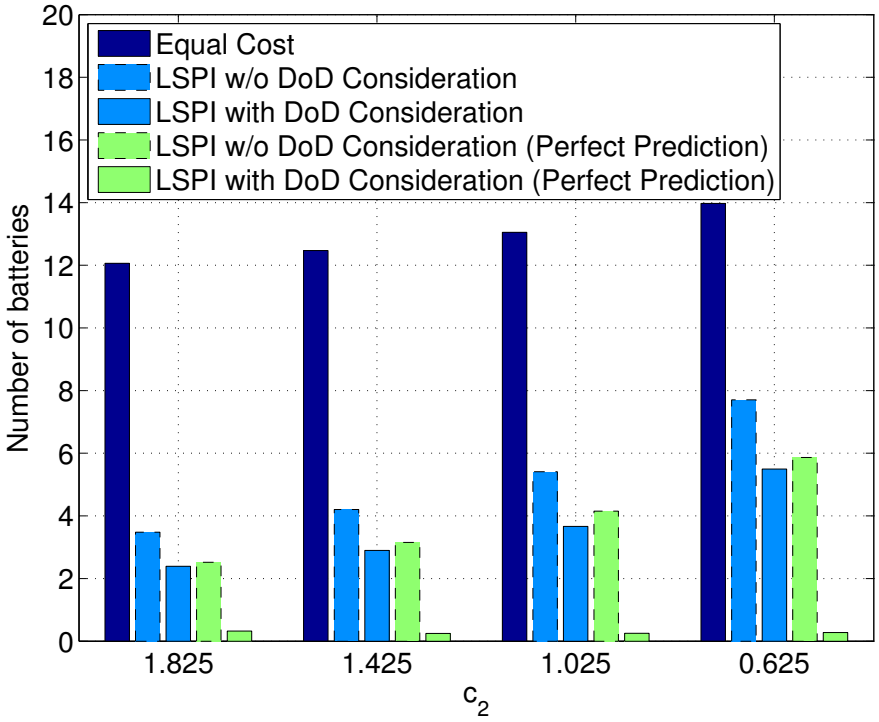

Figure 5.9: Number of battery replacements versus different values of c_2.

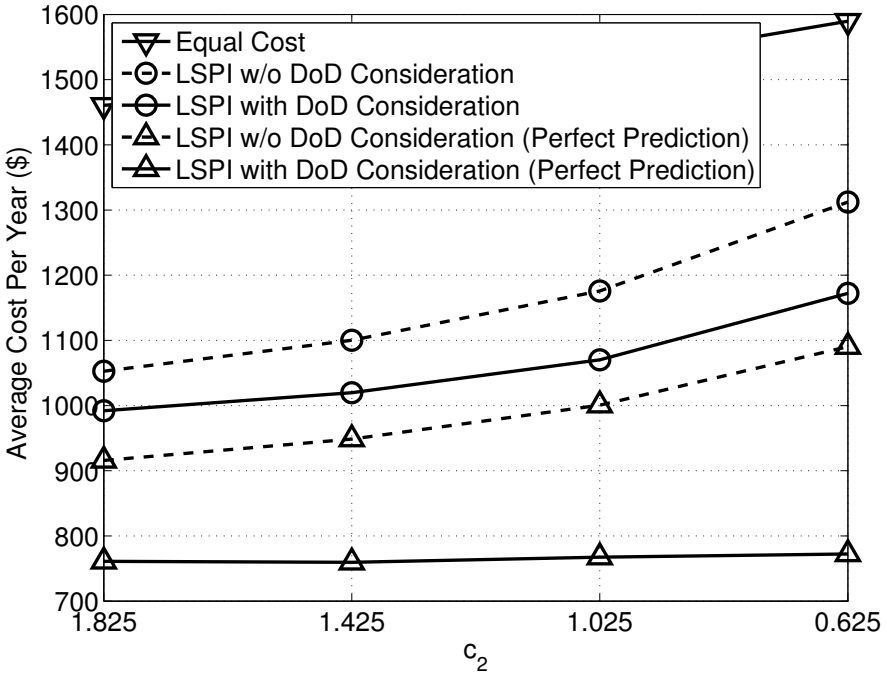

Figure 5.10: Average cost per year versus different values of c_2.

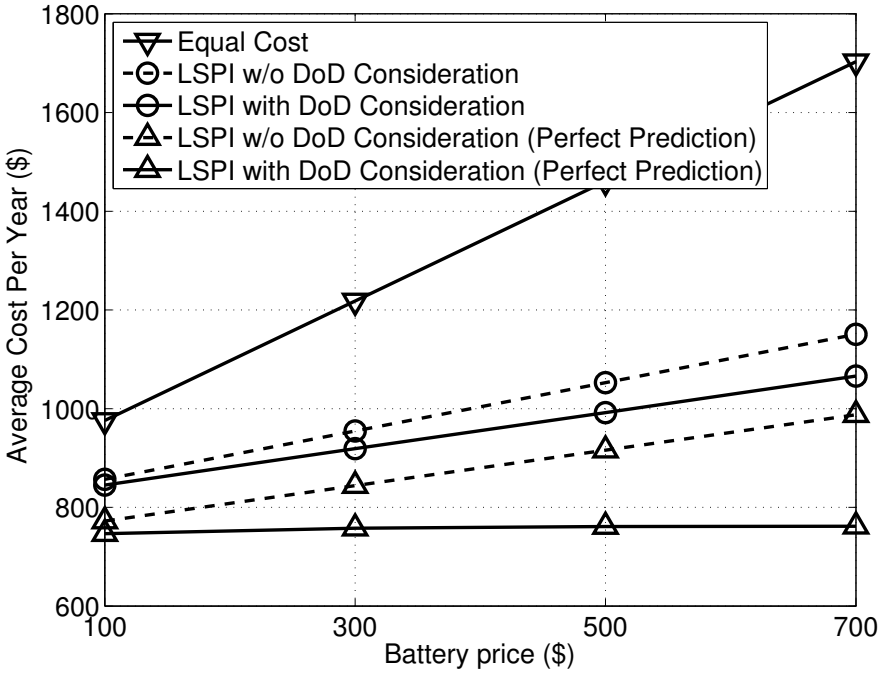

Figure 5.11: Average cost per year versus different values of battery price p_{batt}.

Chapter 6

Conclusion

In this work, we examined a learning-based energy management policy that takes into consideration the tradeoff between the depth-of-discharge (DoD) and the lifetime of batteries and minimize the total cost. First, we proposed a novel battery cost evaluation method that takes into consideration the DoD of each battery usage. We also define the current DoD to describe batteries effect on the marginal cost per battery usage. Second, we utilized the battery cost evaluation method to devise the day-ahead energy management system using reinforcement learning and linear value-function approximations. We transformed energy management system problem into Markov decision process and defined the state transition in our system model. So, we can utilized reinforcement learning algorithm to find the energy purchase policy that minimize the marginal cost. In the end of this work, we conclude that the DoD consideration can help learning algorithm to decide policy that prolong battery life.

Bibliography

[1] Carol Yeager, Betty Hurley-Dasgupta, and Catherine A Bliss. cmoocs and global learning: An authentic alternative. *Journal of Asynchronous Learning Networks*, 17(2):133–147, 2013.

[2] Andrew Ng and Daphne Koller. Coursera. Retrieved May 15, 2016, from the World Wide Web: `https://zh-tw.coursera.org/`, 2012.

[3] Massachusetts Institute of Technology and Harvard University. edx. Retrieved May 15, 2016, from the World Wide Web: `https://www.edx.org/`, 2012.

[4] Mike Sokolsky Sebastian Thrun, David Stavens. Udacity. Retrieved May 15, 2016, from the World Wide Web: `https://www.udacity.com/`, 2012.

[5] Lori Breslow, David E Pritchard, Jennifer DeBoer, Glenda S Stump, Andrew D Ho, and Daniel T Seaton. Studying learning in the worldwide classroom: Research into edx's first mooc. *Research & Practice in Assessment*, 8, 2013.

[6] Daniel T Seaton, Yoav Bergner, Isaac Chuang, Piotr Mitros, and David E Pritchard. Who does what in a massive open online course? *Communications of the ACM*, 57(4):58–65, 2014.

[7] Juho Kim, Phu Tran Nguyen, Sarah Weir, Philip J Guo, Robert C Miller, and Krzysztof Z Gajos. Crowdsourcing step-by-step information extraction

to enhance existing how-to videos. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 4017–4026. ACM, 2014.

[8] Akshay Agrawal, Jagadish Venkatraman, Shane Leonard, and Andreas Paepcke. Youedu: Addressing confusion in mooc discussion forums by recommending instructional video clips. 2015.