Efficient Multi-Level Modeling and Monitoring of End-use Energy Profile in Commercial Buildings

Costas J. Spanos
Zhaoyi Kang

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2015-217
http://www.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-217.html

December 1, 2015

Acknowledgement

This research is funded by the Republic of Singapore's National Research Foundation through a grant to the Berkeley Education Alliance for Research in Singapore (BEARS) for the Singapore-Berkeley Building Efficiency and Sustainability in the Tropics (SinBerBEST) Program. BEARS has been established by the University of California, Berkeley as a center for intellectual excellence in research and education in Singapore.

Efficient Multi-Level Modeling and Monitoring of End-use Energy Profile in Commercial Buildings

by

Zhaoyi Kang

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering - Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Costas J Spanos, Chair
Professor Alexandre Bayen
Professor Stefano Schiavon

Spring 2015

Efficient Multi-Level Modeling and Monitoring of End-use Energy Profile in Commercial Buildings

Abstract

Efficient Multi-Level Modeling and Monitoring of End-use Energy Profile in Commercial Buildings

by

Zhaoyi Kang

Doctor of Philosophy in Engineering - Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Costas J Spanos, Chair

In this work, *modeling* and *monitoring* of end-use power consumption in commercial buildings are investigated through both *Top-Down* and *Bottom-Up* approaches. In the *Top-Down* approach, an adaptive support vector regression (ASVR) model is developed to accommodate the nonlinearity and nonstationarity of the macro-level time series, thus providing a framework for the modeling and diagnosis of end-use power consumption. In the *Bottom-Up* approach, an appliance-data-driven stochastic model is built to predict each end-use sector of a commercial building. Power disaggregation is studied as a technique to facilitate *Bottom-Up* prediction. In *Bottom-Up* monitoring and diagnostic detection, a new dimensionality reduction technique is explored to facilitate the analysis of multivariate binary behavioral signals in building end-uses.

Contents

List of Figures

List of Tables

Acknowledgments

This research is funded by the Republic of Singapore's National Research Foundation through a grant to the Berkeley Education Alliance for Research in Singapore (BEARS) for the Singapore-Berkeley Building Efficiency and Sustainability in the Tropics (SinBerBEST) Program. BEARS has been established by the University of California, Berkeley as a center for intellectual excellence in research and education in Singapore.

Chapter 1

Introduction

1.1 Motivation

In the United States, buildings, both in commercial and residential sectors[1] account for around 40% of the total energy consumption (Figure 1.1), 73% of the total electricity consumption, and 47% of the total natural gas consumption, as illustrated in the EIA's annual energy outlook [23]. Buildings indeed play increasingly important roles in addressing the current energy and climate issues [11]. Significant research & development efforts have been invested in this field of study, such as in the area of control, monitoring, diagnosis, demand response, and more [39][45][73][55][56][40].

Recently, commercial buildings[2], in particular, are drawing more attentions. On one hand, they are usually the dominant consumers of energy and other utilities while being major contributors to increasing energy demands [23]; on the other hand, they employ sophisticated power supply and distribution systems, which enables effective demand side management [79][66][77].

In studying these buildings, it is important to understand their end-use profiles. A building end-use profile aims to evaluate the power consumption of each end-use category in an entire building, for example, space heating, space/room lighting, miscellaneous plug-in loads, shared loads, gas consumption, etc.

Among the many reasons to study the building end-use profile, the first and foremost is the need to better estimate and detect the building's power load and its

[1] industry buildings consume approximately 32% of the total energy consumption, but are not usually included in building energy analysis because of their strong dependence on the related industry activities.

[2] Commercial buildings are defined as buildings with more than half of its floor space allocated for commercial activities, e.g. offices, malls, retail stores, educational facilities, hotels, hospitals, restaurants, etc.

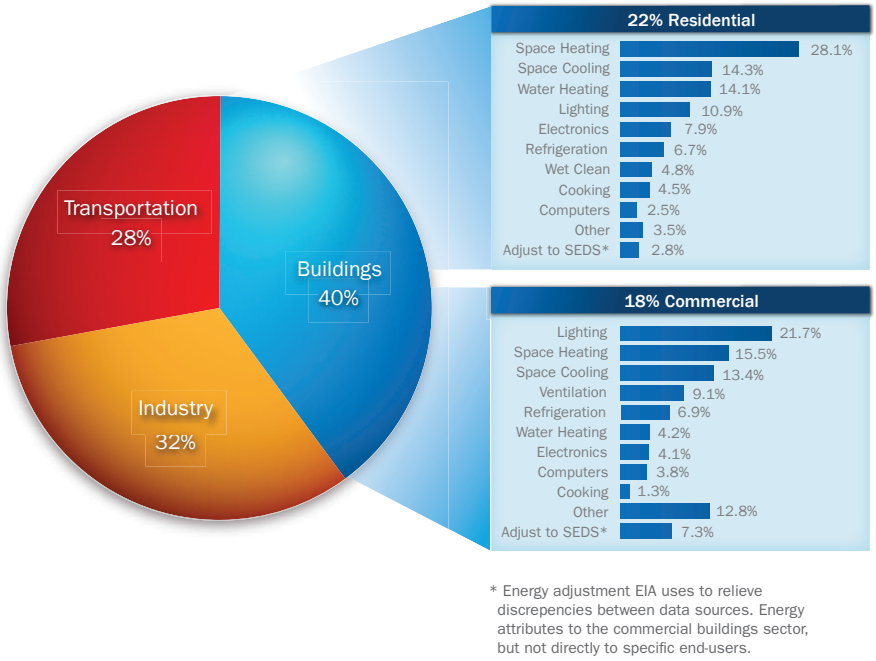

Figure 1.1: Buildings, including commercial and residential sectors, are major contributor to US energy consumption. (*source: Quadrennial Technology Review 2011, US Department of Energy* [19])

variability, so as to better define future requirements in terms of the power plants and the whole power distribution network. The occupant-related load is of special importance in considering the design and evaluation of the *smart* power demand system [67][75][77][62].

Second reason is the requirement of demand side regulation and management. As in the EIA annual energy outlook report [23], to have a 0.6% annual growth in energy consumption (as compared to a residential sector growth of 0.2%), while average floor space increases at a rate of only 1.0% annually. To respond to this rising demand, new demand-response strategies are implemented and Renewable Energy Resources (RERs) are deployed [67][29][79]. Reasonable estimation and diagnosis of the performance in each end-use sector will be required.

The two core components in studying the building end-use profiles are *Modeling* and *Monitoring*. The former involves reasonable prediction or modeling of the end-use consumption, while the latter deals with monitoring and diagnosis of each end-use system. Most of the effort in understanding building end-use profiles will be put into

these two components, as we will show in the next few chapters.

1.2 Two Approaches: *Top-Down* vs. *Bottom-Up*

Building end-use profiles can be usually studied from two perspectives, either *Top-Down* approaches and *Bottom-Up* approaches, distinguished by how the data interact within the approaches [67][29], as illustrated in Figure 1.2:

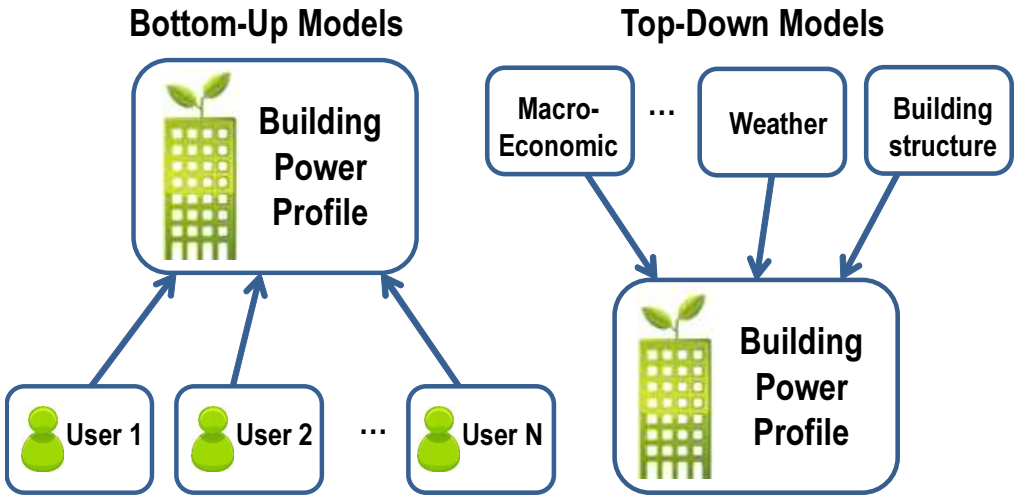

Figure 1.2: Two types of approaches to study the building end-use profiles: Top-Down and Bottom-Up

- The *Top-Down* approach treats a building as a black box and focuses on the *collective demand* of each end-use sector. Usually, a statistical model is built to describe demand variability and used to evaluate the performance of a building's power system. The model would include *macro*-scale extraneous variables, such as macroeconomic indicators (gross domestic product [GDP], income, and price rate), climate, building construction, etc. [29]. Model parameters are estimated from a training set, and building end-use can be modeled or monitored based upon those parameters.

- The *Bottom-Up* aapproach takes into account the individual components in each end-use sector. From the modeling perspective, individual behaviors

can be characterized as a stochastic model, and the whole power consumption can be estimated from Monte Carlo (MC) simulation. The parameters of the stochastic model are estimated from Time-Of-Use (TOU) survey data, which records daily personal usage patterns of each appliance category. From the *monitoring* perspective, instead of whole-building power performance, we are looking at a multivariate occupant-level signal, which contains behavioral information of the building occupants.

Among these two types of approaches, *Top-Down* ones are less complicated and better studied, whereas *Bottom-Up* approaches are relatively new but more adaptive to different scenarios, especially in recent years when building end-use interacts more with occupant behavior through demand-side management. The occupant-dependent fluctuation in power consumption is also directly captured by a *Bottom-Up* approach. On the contrary, *Top-Down* approaches do not typically have the flexibility to do that. In addition, the *Bottom-Up* approach better adapts to changes in the building infrastructure, such as new technologies and new policies, whereas the *Top-Down* approach relies mainly on historical data, as will be illustrated in Chapter 2.

Overall speaking, *Bottom-Up* approach will be more thoroughly studied in this work, while a few issues about the *Top-Down* approach will also be addressed.

1.3 Current Challenges

Challenges in *Top-Down* approach

Statistical modeling is at the core of any *Top-Down* approach for both monitoring and modeling purposes. Most current models, however, especially the linear Gaussian random noise statistical models, have limited capability to handle deviations from linearity or stationarity, which is often observed in building end-use profiles.

Challenges in *Bottom-Up* approach

Bottom-Up analysis of building energy has been a difficult task, since measuring each end-use category is costly. In recent years, this problem is easier to tackle, thanks to the development of large-scale wireless sensor networks and distributed data storage systems. Many existing works have demonstrated such a development, such as in[39], [38], [45], etc. However, several issues still need to be addressed.

From the *modeling* perspective, behavior-dependent end-use sectors, such as plug-in loads, occupant-controlled lighting, and occupant-adjusted HVAC, have a significant amount of diversity and fluctuation [39] while being the bottlenecks to demand-

side management [57]. Better capture of this variance in a new model is highly preferred.

From the *monitoring* perspective, on one hand, measuring end-use *bottom*-level power consumption brings up several issues. More specifically, deploying sensors to each appliance in modern commercial buildings will be costly, while this method also introduces privacy issues. Therefore, a model objective should include low density, non-intrusive monitoring. On the other hand, monitoring, and even diagnosis or control, will be challenging in larger buildings if there are a great amount of individual appliances. In fact, from both a statistical and engineering perspective, a meticulous analysis will be wasteful. A concise but reliable description of the appliances in each end-use category is preferred.

This thesis focuses on issues described above while demonstrating potential solutions for both *modeling* perspective and *monitoring* perspective.

1.4 Thesis Outline

The rest of the thesis is organized as follows:

Chapter 2 presents a non-parametric statistical model, adaptive support vector regression (ASVR), as a *Top-Down* approach to address non-linearity and non-stationarity issues in the *macro*-scale modeling of commercial building end-uses.

Chapter 3 moves from *macro*-level *Top-Down* approaches to *micro*-level *Bottom-Up* approaches. We demonstrate a *Bottom-Up* appliance-data-driven stochastic ON/OFF probability model is demonstrated to stochastically estimate the end-use of different categories of appliances followed by a discussion about a non-homogeneous Poisson process approach to model the shared appliances.

In Chapter 4, based on the study in Chapter 3, a *Bottom-Up* approach is used to model real building plug-in loads power consumption under different scenarios. A new power disaggregation technique is proposed, which is used to filter out ON/OFF states of individual appliances from aggregated raw power stream.

In Chapter 5, challenges in *Bottom-Up* monitoring are addressed. A dimensionality reduction technique, Logistic PCA (LPCA), is deployed to deal with binary behavioral data in the *Bottom-Up* perspective, and a sequential version of Logistic PCA (SLPCA) is proposed and analyzed.

Finally, Chapter 6 concludes this study and includes a brief discussion about future tasks found within the topics studied by this thesis.

Chapter 2

Top-Down Approach for End-Use Modeling & Monitoring

2.1 Introduction

In this chapter, a *Top-Down* approach is discussed, followed by the proposed non-parametric adaptive support vector regression method for *macro*-level building end-use modeling.

As illustrated in Chapter 1, a *Top-Down* approach treats the building as a black box and uses historical data or other physical parameters as features to build up a statistical model. The widely used features include historical building power consumption for each sector; physical parameters, such as the construction area, material, structures, etc.; environmental parameters, such as the temperature, humidity, sunlight level, rain precipitation, etc.; and *macro*-economic features, such as gross domestic product (GDP), salary level, unemployment rate, appliance penetration level, etc.

Model prediction capability is critical. Researchers have studied two types of models, namely, physical and statistical. Physical models simulate the energy consumption from thermodynamics standpoint. Examples of this approach include EnergyPlus[1], which is a software developed by the Building Technology Office of the US Department of Energy. A physical model usually gives accurate results and can be more adaptive to the change of the building structure and material. However, calculation is usually too tedious to be used in a real-time monitoring and evaluation platform.

On the other hand, statistical models are empirical in nature (i.e., based on

[1] http://apps1.eere.energy.gov/buildings/energyplus/

observation) and are usually implemented as linear or nonlinear regression on a set of features. The features can be selected based on statistical significance rather than on physical principles. Future power consumption is usually extrapolated from the features. Statistical models are usually too simple to provide highly accurate results, but they are statistically robust and computationally efficient. Hence, they are preferred in real-time modeling and monitoring.

In this chapter, we will focus on statistical methods and develop an adaptive least-mean square version of the nonlinear time series model, which could be used in real-time building end-use monitoring and diagnosis.

The rest of this chapter is organized as follows: Section 2.2 presents a literature review on existing models and challenges. Section 2.3 introduces the linear autoregressive model. Section 2.4 briefly talks about challenges lying in the current data feed. Section 2.5 discusses an adaptive support vector regression model. Section 2.6 gives results and discussion, while Section 2.7 concludes with discussions about future tasks.

2.2 Prior Works

Prior *Top-down* studies apply physical, statistical, or econometric models to use historical data or other features to predict load curve e [78].

Physical models have been developed as software tools, such as DOE-2[2], EnergyPlus[3], BLAST[4], ESP-r[5]. An overview can be found in [14], and an updating list of these tools can be found in [18]. These tools, in most cases, use very detailed information about the building, which becomes time-consuming in both training and estimation.

Statistical methods have been developed as approximate but computationally efficient alternatives. There is an extensive amount of work on these topics, including linear regression methods developed for different geographical or climatic conditions [46], linear time series models, the so-called Conditional Demand Analysis (CDA) [2], the Back Propagation Neural Network (BPNN) based methods [35], and the Support Vector Machine (SVM) [47]. The linear regression or time series model take the least amount of parameters, whereas BPNN could take more complicated model structure [78].

[2] http://doe2.com/DOE2/

[3] http://apps1.eere.energy.gov/buildings/energyplus/

[4] http://apps1.eere.energy.gov/buildings/energyplus/energyplus_research_legacy.cfm, EnergyPlus is actually a merge of DOE-2 and BLAST

[5] http://www.esru.strath.ac.uk/Programs/ESP-r.htm

SVM [13] based methods are attracting more and more attentions recently because of their flexibility to model nonlinear behaviors and their relatively acceptable model complexity. In time series modeling, it can be extended to Support Vector Regression (SVR) with Auto-regressive terms [65]. Several works have done on exploring the application of this model [65] [51] [49] [24].

In this chapter, we will study an adaptive autoregressive SVR model for nonlinear time series that can be used effectively to estimate energy consumption with great extendibility.

2.3 Linear Auto-regressive Model

Conventionally, the linear auto-regressive model has been used to model time series data. Some well-known methods include the Auto-Regressive (AR) model, the Auto-Regressive Moving-Average (ARMA) model, and more. The AR model gives an estimation of a certain data point based on a linear extrapolation of its own history.

As an example, for time series $x_1, \cdots, x_n \doteq \{x_t\}_{t=1}^{T}$, we model x_t based on a weighted sum of x_{t-1} through x_{t-q}.

$$x_t = \sum_{i=1}^{q} \beta_i x_{t-i} \tag{2.1}$$

in which q is the order of the AR model and $\{\beta_i\}_{i=1}^{q}$ are the parameters. The parameters can be learned from minimizing the sum of square error as:

$$\hat{\beta}_i, \cdots, \hat{\beta}_q = \underset{\beta_1, \cdots, \beta_q}{\operatorname{argmin}} \sum_t \left(x_t - \sum_{i=1}^{q} \beta_i x_{t-i} \right)^2 \tag{2.2}$$

By writing $\beta_0 = -1$, we can transform equation (2.1) as:

$$\sum_{i=0}^{q} \beta_i x_{t-i} = 0$$

Here we introduce the *Backward-operator* as $\mathcal{B}_i x_t \doteq x_{t-i}$, and hence:

$$\sum_{i=0}^{q} \beta_i x_{t-i} = \sum_{i=0}^{q} \beta_i \mathcal{B}_i x_t = \left(\sum_{i=0}^{\tau} \mathcal{B}_i \right) x_t = \phi_q(\mathcal{B}) x_t = 0$$

In which $\phi_q(\mathcal{B}) = \sum_{i=0}^{\tau} \beta_i \mathcal{B}_i$. Similarly, seasonality can be easily added by Seasonal-AR (SAR) model.

$$\sum_{j=0}^{s} \eta_j \phi_q(\mathcal{B}) x_{t-j \cdot S} = \sum_{j=0}^{s} \eta_j \mathcal{B}_j^S \phi_q(\mathcal{B}) x_t = \phi_s(\mathcal{B}^S) \phi_q(\mathcal{B}) x_t = 0 \tag{2.3}$$

in which s is the period and $\phi_s(\mathcal{B}^S) = \sum_{j=0}^{s} \eta_j \mathcal{B}_j^S$.

An important assumption of these kind of models is that the model parameter set $\{\beta_i\}_{i=1}^{q}$ or $\{\eta_j\}_{j=1}^{s}$ is stationary, which means the parameters are invariant over time. If the parameters are subject to change, linear AR or SAR model will not be able to capture that.

2.4 Challenges in Linear Auto-regressive Model

The linear auto-regressive models, although delivering well-formed theory, are subject to practical issues.

- Usually, AR, especially the SAR model, needs a large amount of data for model training.
- Additionally, linear AR or SAR models are not suitable for nonlinearity application.
- Furthermore, as illustrated earlier, linear AR or SAR modeling assumes stationarity of the model. If the data is nonstationary, then linear modeling is not enough.

Take B-90 building in Lawrence Berkeley National Laboratory of U.S. Department of Energy (DOE) as an example (as shown in Figure 2.1). Two building-level power consumption time series of B-90 are shown in Figure 2.2. The data are measured by DENT meter[6] (as in Figure 2.3) in one hour or 15 min intervals, and collected through sMAP protal[7].

In Figure 2.2, a strong periodic pattern and chaotic glitches can be observed. Modeling by simple AR-type models is not enough. Recently, non-parametric data-driven methods have been proposed to overcome this issue. In this work, we will study an alternative model, which is called the Adaptive Support Vector Regression (ASVR) model.

[6] http://www.dentinstruments.com
[7] http://new.openbms.org/plot/

Figure 2.1: Building-90 in Lawrence Berkeley National Laboratory (LBNL)

2.5 Adaptive Support Vector Regression

In ASVR model, modeling problem is formed in a different way. Let $\mathbf{u}_t = \{x_{t-1}, \cdots, x_{t-r}\}$ be the autoregressive term, and our estimation of a certain data point x_t would be $\beta^T \mathbf{u}_t$. By introducing a soft error bound of the difference $x_t - \beta^T \mathbf{u}_t$ similar to the famous soft margin Support Vector Machine (SVM) [13] [69], we can put the optimization problem (2.2) in the following form.

$$\begin{aligned} \min_{\beta_i, \forall i, \xi_t^+, \xi_t^-, \forall t} \quad & \sum_{t=1}^{n} \left(\xi_t^+ + \xi_t^-\right)^2 + \frac{1}{2}\|\beta\|^2 \\ \text{s.t.} \quad & -\xi_t^- \leq x_t - \beta^T \mathbf{u}_t \leq \xi_t^+, \forall t \end{aligned} \tag{2.4}$$

The $\frac{1}{2}\|\beta\|^2$ is a regularization term, indicating that a flat or small β is preferred here. (2.4) is a convex optimization problem. The Lagrangian function is as below [65]

$$\begin{aligned} L(\alpha_t^+, \alpha_t^-, \xi_t^+, \xi_t^-) = & \sum_{t=1}^{n} \left(\xi_t^+ + \xi_t^-\right)^2 + \frac{1}{2}\|\beta\|^2 \\ & + \sum_{t=1}^{n} \alpha_t^+ (x_t - \beta^T \mathbf{u}_t - \xi_t^+) + \sum_{t=1}^{n} \alpha_t^- (-\xi_t^- - x_t + \beta^T \mathbf{u}_t) \end{aligned} \tag{2.5}$$

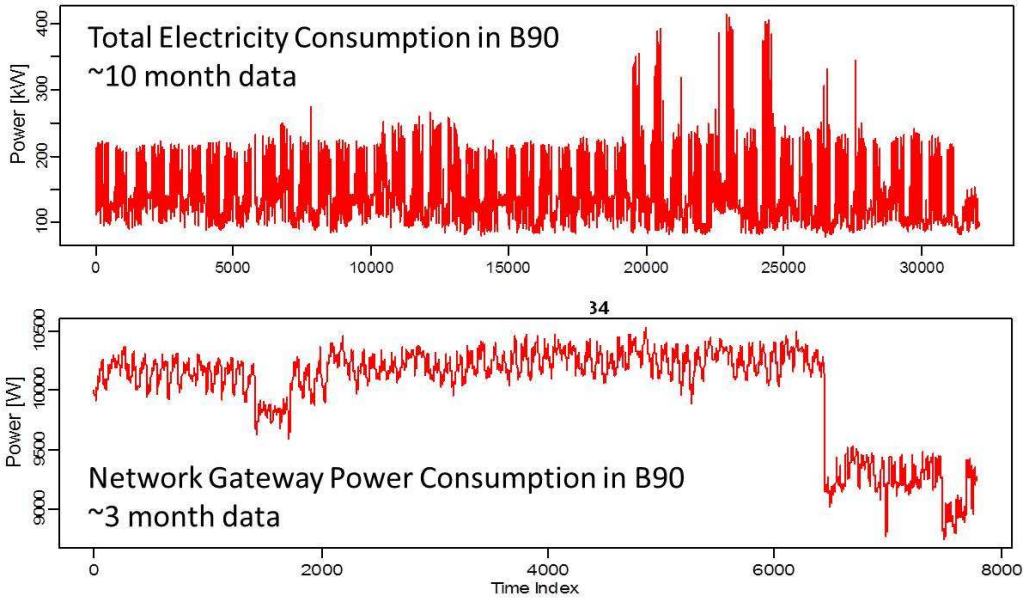

Figure 2.2: Example of data collected from Building-90. Total Electricity consumption (upper) and Network gateway node consumption (bottom).

in which α_t^+, α_t^- are the positive Lagrangian multipliers. Following the KKT condition of this strictly convex problem [8], we have the following conditions:

- Derivative v.s. β:

$$\frac{\partial L}{\partial \beta} = \beta - \sum_{t=1}^{n} \alpha_t^+ \mathbf{u}_t + \sum_{t=1}^{n} \alpha_t^- \mathbf{u}_t = 0 \tag{2.6}$$

- Derivative v.s. $\xi_t^+, \forall t$:

$$\frac{\partial L}{\partial \xi_t^+} = 2 \sum_{t=1}^{n} (\xi_t^+ + \xi_t^-) - \sum_{t=1}^{n} \alpha_t^+$$

- Derivative v.s. $\xi_t^-, \forall t$:

$$\frac{\partial L}{\partial \xi_t^-} = 2 \sum_{t=1}^{n} (\xi_t^+ + \xi_t^-) - \sum_{t=1}^{n} \alpha_t^-$$

Figure 2.3: Dent meter used to collect the macro-level data

- Complementary slackness, $\forall t$:

$$\alpha_t^+(x_t - \beta^T \mathbf{u}_t - \xi_t^+) = \alpha_t^-(-\xi_t^- - x_t + \beta^T \mathbf{u}_t) = 0 \tag{2.7}$$

The Equation (2.6) satisfies:

$$\hat{\beta} = \sum_{i=1}^{n} (\alpha_i^+ - \alpha_i^-)\mathbf{u}_i = \sum_{i=1}^{n} \alpha_i \mathbf{u}_i$$

Notice that here the α_i terms do not need to be positive. Hence, the estimated observation follows:

$$\hat{x}_t = \hat{\beta}^T \mathbf{u}_t = \sum_{i=1}^{n} \alpha_i \langle \mathbf{u}_i, \mathbf{u}_i \rangle \tag{2.8}$$

Due to (2.7), only part of the α_i's are non-zero, corresponding to points with equality in the constraints in (2.4). Besides, following equation (2.8), only those data points contribute to the weighted sum of estimation. In Figure 2.4, we can see that the circled dots are those corresponding to the data with non-zero α_i's. Those data points are called Support Vectors (SVs), in that they *support* the shape of the curve.

In case of nonlinearity, in the observed data in Figure 2.2. An alternative way is to map the input $\mathbf{u}_t$ into another domain $\phi(\mathbf{u}_t)$ in which the relationship is linear, we have:

$$x_t = \sum_{i=1}^{n} \alpha_i \langle \phi(\mathbf{u}_t), \phi(\mathbf{u}_i) \rangle = \sum_{i=1}^{n} \alpha_i k(\mathbf{u}_t, \mathbf{u}_i) \tag{2.9}$$

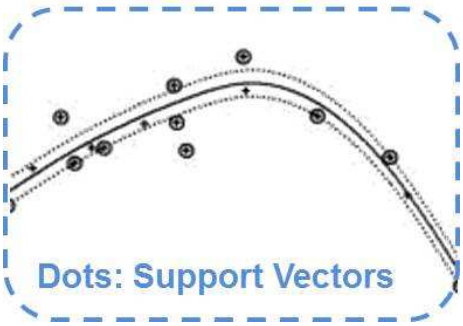

Figure 2.4: Support vectors

in which we have $k(\mathbf{u}_t, \mathbf{u}_i) = \langle \phi(\mathbf{u}_t), \phi(\mathbf{u}_i) \rangle$, called the Kernel function. In (2.9), we don't really need to know the form of $\phi(\cdot)$, as long as we have an idea about the Kernel function $k(\cdot, \cdot)$. This is also called the kernel trick [69].

The most widely used kernel function is the Gaussian kernel, which can be written in the form of $k(\mathbf{u}_t, \mathbf{u}_i) = e^{-\sigma \|\mathbf{u}_t - \mathbf{u}_i\|^2}$. Gaussian kernel quantifies the correlation or similarity between $\mathbf{u}_t, \mathbf{u}_i$. Other widely used kernel function includes the polynomial kernel function $k(\mathbf{u}_t, \mathbf{u}_i) = \|\mathbf{u}_t - \mathbf{u}_i\|^p$.

When the data is in real time, it is costly to form a convex optimization problem as (2.4) at every step. A solution to this is to put it in a recursive least square formulation. For RLS, we can learn α_i's in Equation (2.9) recursively. Due to the complementary slackness in (2.7), some data points may contribute to the shape of the curve, and some data points may not be support vectors.

Since support vectors are those data points critically determines the shape of the curve, they usually demonstrates less similarity compared to the previous data points. Hence, we can determine whether a data point is support vectors by examining the kernel function $k(\mathbf{u}_t, \mathbf{u}_i)$ between $\mathbf{u}_t$ to all the previous $\mathbf{u}_i$'s, as well as examining the error of estimation $x_t - \sum_{i=1}^{t-1} \alpha_i k(\mathbf{u}_t, \mathbf{u}_i) = \sum_{i \in \text{SVs}} \alpha_i k(\mathbf{u}_t, \mathbf{u}_i)$ in which SVs is the support vector dictionary[8], following the idea of [42], [24], [49] and [61].

- Let $K_t = [\cdots, k(\mathbf{u}_t, \mathbf{u}_i), \cdots], \forall i \in \text{SVs}$.
- For each time step $t = 1, \cdots, n$, we have the error term $\text{Err}_t = x_t - K_t^T \alpha_t$, and a distance with respect to the kernel functions $\text{Dist}_t = \max_{j \in \text{SVs}} \|k(\mathbf{u}_t, \mathbf{u}_i)\|$

[8]By support vector dictionary, we mean the collection of all the support vectors up to the current data point

- If $\text{Err}_t \leq \mu$ and $\text{Dist}_t \leq \varpi$. Let $\tilde{K}_t = [K_t^T, 1]^T$ and $\tilde{\alpha}_t = [\alpha_t^T, 0]^T$, update the coefficient as:

$$\tilde{\alpha}_{t+1} = \tilde{\alpha}_t + \eta \frac{x_t - \tilde{K}_t^T \tilde{\alpha}_t}{\|\tilde{K}_t\|^2 + \rho} \tag{2.10}$$

  in which η is the learning rate. The larger the η, the more adaptive to the change in the process.

- Else, update the coefficient as

$$\alpha_{t+1} = \alpha_t + \eta \frac{x_t - K_t^T \alpha_t}{\|K_t\|^2 + \rho} \tag{2.11}$$

Therefore, we learn the parameters α_i's. When a new data point is determined as a support vector, we just add it into the support vector dictionary, and change the dimensionality of α accordingly.

2.6 Results and Discussion

Firstly, we examine the online evolution of the support vector dictionary. As shown in Figure 2.7, we have several key observations.

- Usually, only 15% of the data points are support vectors, which means we only need to store a small portion of the data but are able to capture most of the fluctuation, nonlinearity, and nonstationarity of the time series.

- The support vectors mostly appear around change-points or nonlinear patterns of a time series, exactly as expected.

- There is definitely a trade-off between accuracy and the dictionary-size. The more support vectors we have, the more capable we are to capture the original pattern; however, there is more storage cost. The number of support vectors can be tuned by changing the hyperparameter in the kernel function (σ). This is illustrated in Figure 2.5 when running the algorithm on a three-month total plug-in loads power consumption in the CREST center, and an almost-monotonic pattern is observed. Moreover, same as the conventional linear model, an overly detailed model suffers from over-fitting, which is illustrated in Figure 2.6. The total plug-in loads are also used in Figure 2.5, but two-month's data is used as training and one-month's data is used as testing. In Figure 2.6, training error decreases as expected when we have a more detailed model, whereas testing error demonstrates a bowl shape when compared

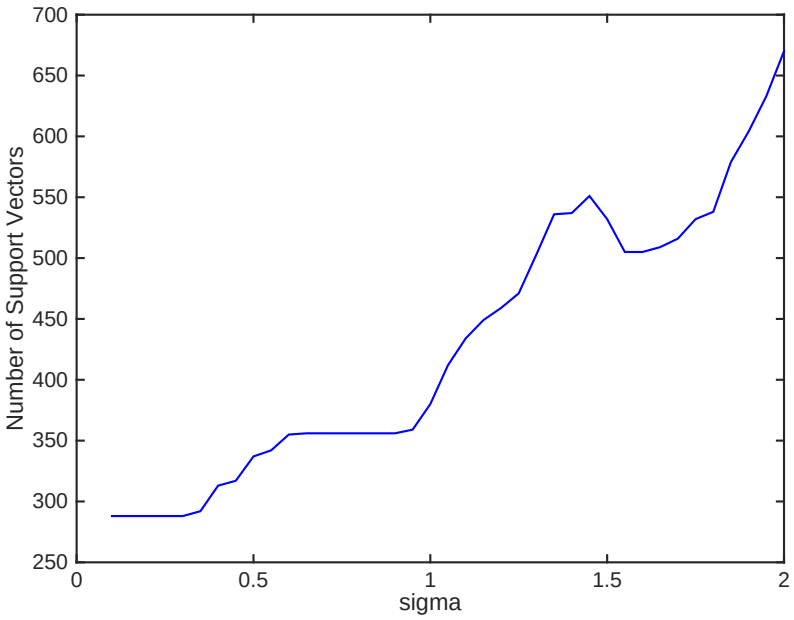

Figure 2.5: Number of support vectors as a function of hyperparameter in kernel function (σ)

to the hyperparameter. In real time, a hyperparameter is most often used that is not too complicated but detailed enough to provide reasonable accuracy.

The ASVR can be used in pattern discovery with great extendibility. Specifically, a data point is added into the support vector dictionary when a new pattern appears, and thus, prediction error or kernel distance function increases, as shown in Figure 2.8.

The distance measurement (in other words, the change recognition functions in Figure 2.8) can also be altered to accommodate different scenarios. For example,

$$\text{Dist}_t = \max_{j \in \text{SVs}} \| k(\mathbf{u}_t, \mathbf{u}_i) \underbrace{\exp\left(-\frac{1}{\alpha} |1 - \cos(\frac{2\pi \Delta t}{\omega})| \right)}_{\text{Periodic weight}} \times \underbrace{\exp\left(-\frac{\delta \Delta t^2}{\omega} \right)}_{\text{Decay}} \| \tag{2.12}$$

It is worth mentioning that the choice of the kernel distance function (or change recognition function) can affect the support vector dictionary's distribution as well, which will be a subject of future work.

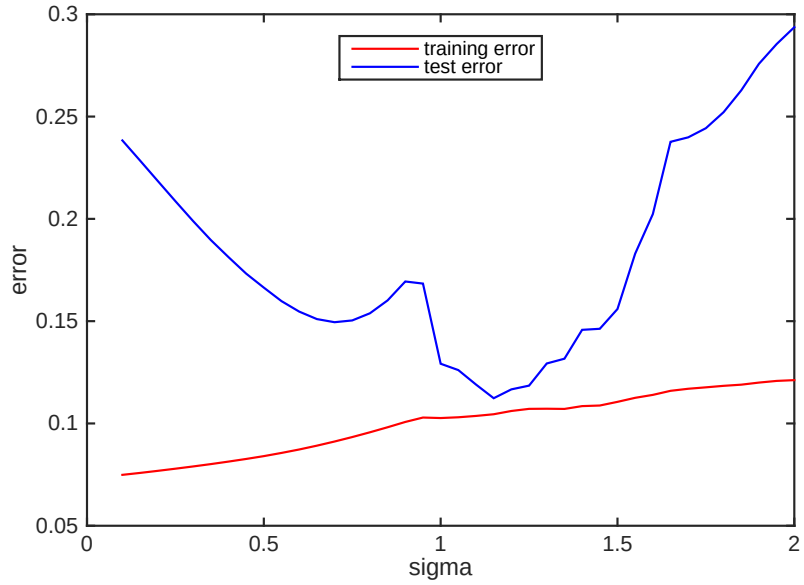

Figure 2.6: Training and Testing error as a function of hyperparameter in kernel function (σ)

2.7 Conclusion and Future Tasks

In this chapter, we discuss the *Top-Down* modeling of building end-use power consumption. The linear auto-regressive model is studied and its limitations in dealing with nonlinearity and nonstationary are discussed. A non-parametric data-driven adaptive support vector regression (ASVR) model is introduced as an alternative approach. The ASVR model can effectively capture nonlinearity and nonstationary by storing only a small portion of the original data points.

The future tasks of this chapter would be the design of proper distance function (or change recognition score function) to cope with different types of nonlinearity or nonstationarity, and the method could be extended to the fault diagnosis problem. Including more parameters into the model will also be useful.

However, it should be noted that this method is a so-called black-box method. It can capture statistically significant features of a building but provides little information about occupant-dependent information, which is, unfortunately, of special importance in modern smart building operation.

In the next several chapters, we will move on to the discussion of *Bottom-Up*

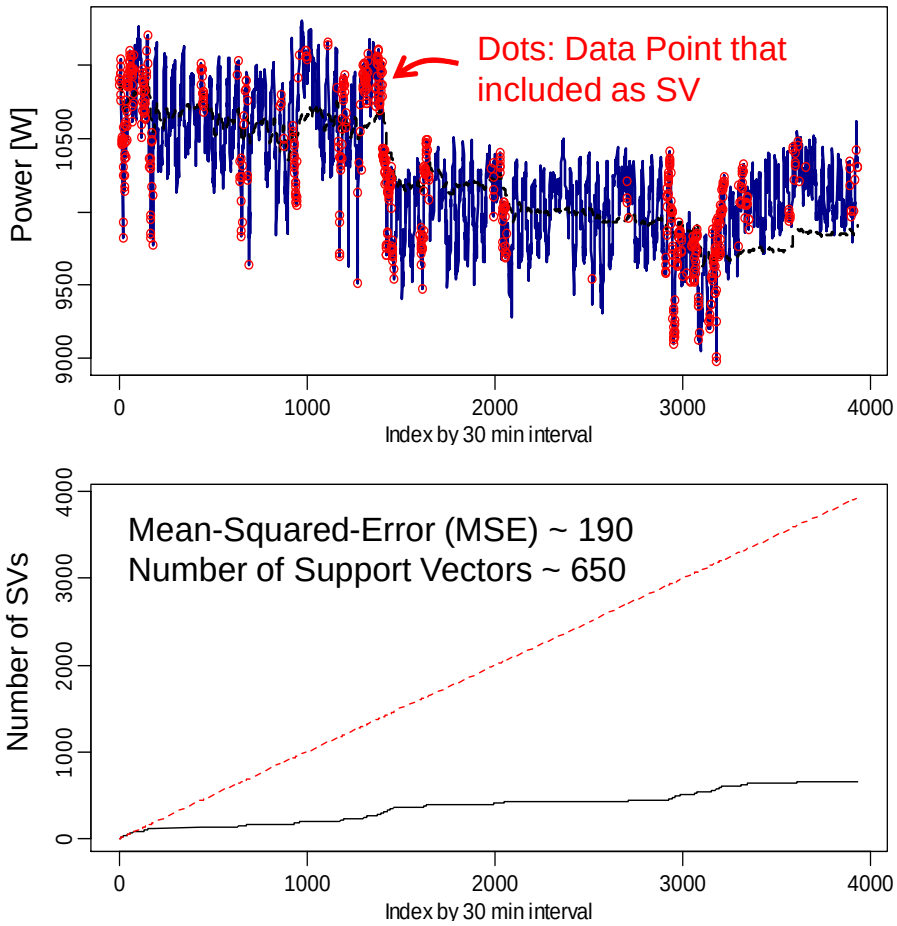

Figure 2.7: Evolution of support vectors dictionary

approaches, which are more capable of modeling occupant-dependent features.

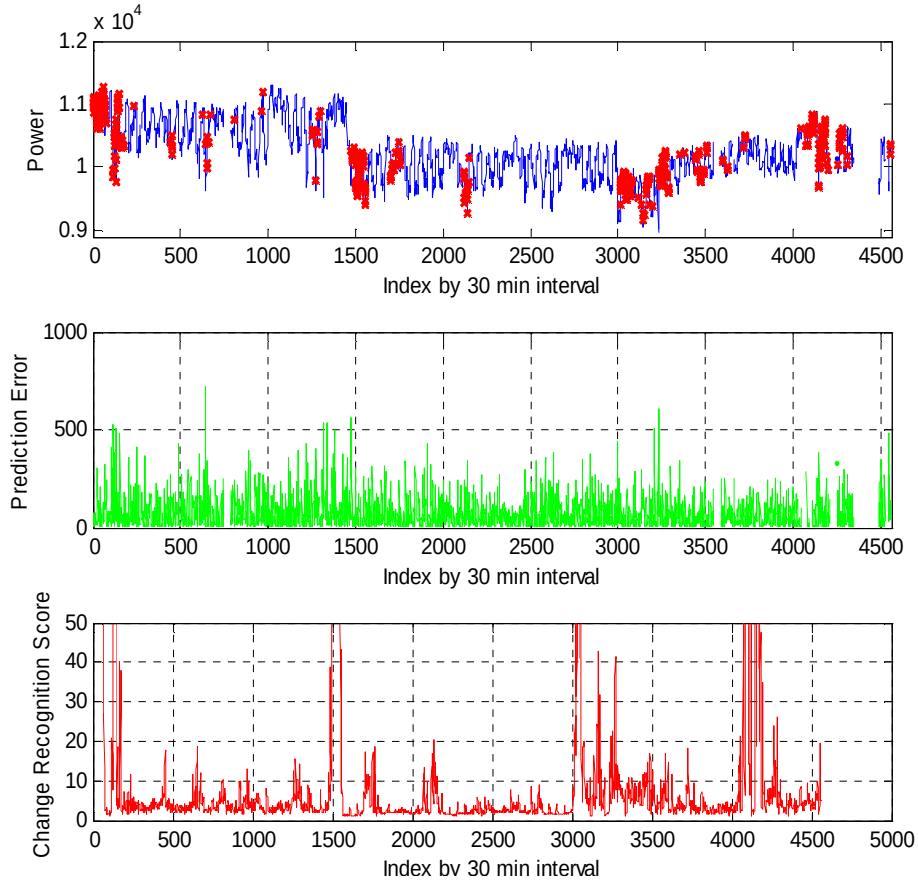

Figure 2.8: Application of pattern recognition of adaptive support vector regression (ASVR). The Prediction Error corresponds to $\text{Err}_t = x_t - K_t^T \alpha_t$; the Change Recognition Score is the distance function.

Chapter 3

Bottom-Up End-Use Modeling: Model Setting

3.1 Background

The last chapter briefly introduced the *Top-Down*approach, and here we will move on to the *Bottom-Up* approach. As mentioned before, *Bottom-Up* approaches are relatively new and attracting more attention in recent years, because of their capability in evaluating occupant demand and its adaptability under different strategic scenarios. In the next two chapters, we will firstly discuss *modeling* issues under *Bottom-Up* settings and then study *Bottom-Up monitoring* issues in Chapter 5.

One of the earliest works on *Bottom-up* models is written by A. Capasso *et al.* [10]. Presence probability is used to model the likelihood that a resident is in a house. Activity probability is used to model how likely it is that an activity will be happening. These probabilities are extracted from Time-Of-Use (TOU) data. TOU data comes from survey recordings of residents' daily activities in 15-min time intervals. Together with duration statistics1 obtained from prior knowledge, a power stream can be generated by Monte Carlo (MC) simulation. In[74], TOU data is used again, and nine synthetic activity patterns are defined. A non-homogeneous Markov Chain is used to model the turn-ON events of each activity. Duration and ON events are sampled randomly from the estimated distribution. In [62], activity probability is also estimated from TOU data and other extraneous data, so that is non-homogeneous. In [75], estimation of activity probability patterns is based on TOU survey, duration statistics, and a more elaborate model.

Existing methods that employ the *Bottom-Up* approach provide great insights into end-use profile models of commercial buildings. However, there are still several

remaining issues:

- Previous works mostly used TOU data to obtain indoor activity probability, and then activity was converted to appliance pattern through an empirical model. This is sometimes problematic, since conversions are usually not rigorously justified.
- In commercial buildings, variation of power consumption among buildings is not of significant interest, since the infrastructures of different buildings can significantly vary, whereas variation among users becomes especially interesting, since it can indicate performance limits of a building's power system. However, the latter is not thoroughly studied in previous work.
- Cross-correlation among appliances is not directly captured in the past. A random Markov Chain model could under-estimate the demand. Moreover, most previous research mentioned modeling shared activities, whereas validation of these models is difficult.

In this chapter, we will directly estimate probability patterns of appliances in commercial buildings and develop a model based on the turning-ON/OFF probability of appliances to quantify the variation of building end-use power profile. We will also address correlation between appliances with a correction term.

This chapter is organized as follows: In Section 3.2, the big picture of the *Bottom-Up* model is discussed. In Section 3.3, the Statistical Parameters in the model are investigated. Sections 3.4 review the models of shared appliances. In Section 3.5, a conclusion is given.

3.2 Big Picture

A *Bottom-up* model can be viewed as a gray-box that takes two types of parameters, as shown in Figure 3.1.

One is called the *Statistical Parameter*, which describes statistical properties of appliances (e.g., ON/OFF probability, presence probability, duration statistics, etc.). This type of parameter is usually extracted from appliance usage data collected by wireless sensor networks, and it can be learned in one building and extended to other buildings with similar profiles. For example, if a model is built for student space, it can be extended to other school buildings.

The other type of parameters is called the *Field Parameter*, which includes the number of occupants, number of computers, monitors, printers, microwaves, etc.,

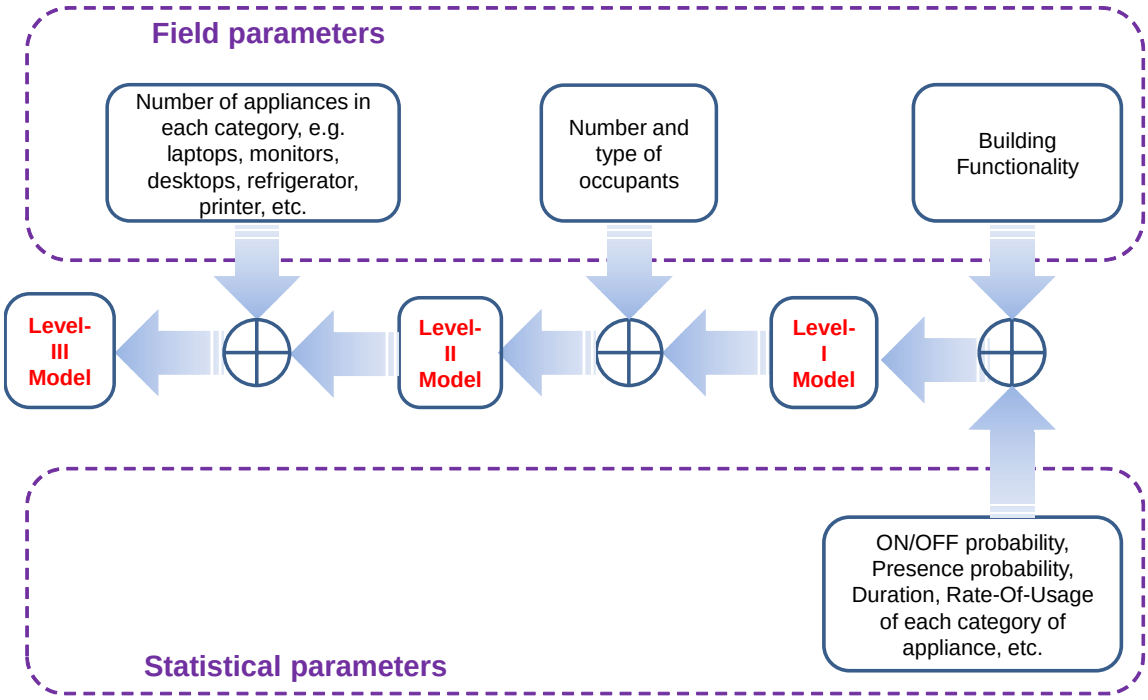

Figure 3.1: Parameters in *Bottom-up* model: Field Parameters and Statistical Parameters. *Level-III* model is the most complex, and *Level-II* model is less complex; *Level-I* is the simplest but low accuracy.

depending on building structure and utility. These parameters are collected from field study or empirical knowledge and will be evaluated in the CREST center, the SWARM lab, and the fourth floor of Sutardja-Dai Hall, all at UC Berkeley, as will be discussed in more detail in Chapter 4.

Based on the complexity of the *Field Information*, we can further divide the models into *Level-I* model, *Level-II* model and *Level-III* model.

- In the most simplified *Field Parameter* setting, we only know the building functionality. Occupant characteristics (e.g., the number of occupants, number of desktop and laptops, etc.) are inferred from building functionality, and we call this kind of model the *Level-I* model. *Level-I* will be most welcomed in commercial application, but its accuracy cannot be guaranteed.

- As we get to know more information of the occupants, for example, the number and type of occupants, we are closer to the appliances, and the accuracy could

be better. The relatively complex model is called the *Level-II* model.

- The *Level-III* model directly contains parameters about the appliances, such as the number of desktops, laptops, monitors, printers, microwaves, lamps, etc. However, though demonstrating great accuracy, these data are relatively costly to collect, or even unavailable, especially for early-stage power system design.

To achieve better accuracy in this work, only the relative more complex models, in other words, *Level-II* and *Level-III* models, are considered.

3.3 Statistical Parameters

Previously, people use different types of statistical parameters in their end-use model. We can roughly divide their methodologies into the following three modules: rate-of-use statistics, duration statistics, and ON/OFF-probability statistics.

To facilitate the analysis, for an appliance, given that we have d days of observations, we define $S_t^{(i)}$ as its state of i-th day, i.e. $S_t^{(i)} \in \{0, 1\}$ and 1 stands for ON.

Rate-of-Use Statistics

Rate-Of-Use (ROU) statistics is a basic model used to describe appliance usage.

Definition 3.3.1 (Rate-Of-Use)**.** *Rate-Of-Use (ROU) is the portion of time that the appliance is ON in each time-of-day:*

$$\mathrm{ROU}_t = \frac{1}{d}\sum_{i=1}^{d} S_t^{(i)} = \overline{S_t} \tag{3.1}$$

For example, in the 80 days of experiment, the monitor is ON at 12:00PM in 16 days, the ROU would be $16/80 = 0.2$ at 12:00PM. The ROU is plotted for monitor, laptop and desktop in Figure 3.2. Strong daily pattern is observed. ROU indicates the average energy consumption, but it doesn't indicate the usage pattern of the appliance.

Duration Statistics

Duration statistics were used to characterize duration time of each activity [62] [75]. We extracted the duration statistics from sensor data after power disaggregation.

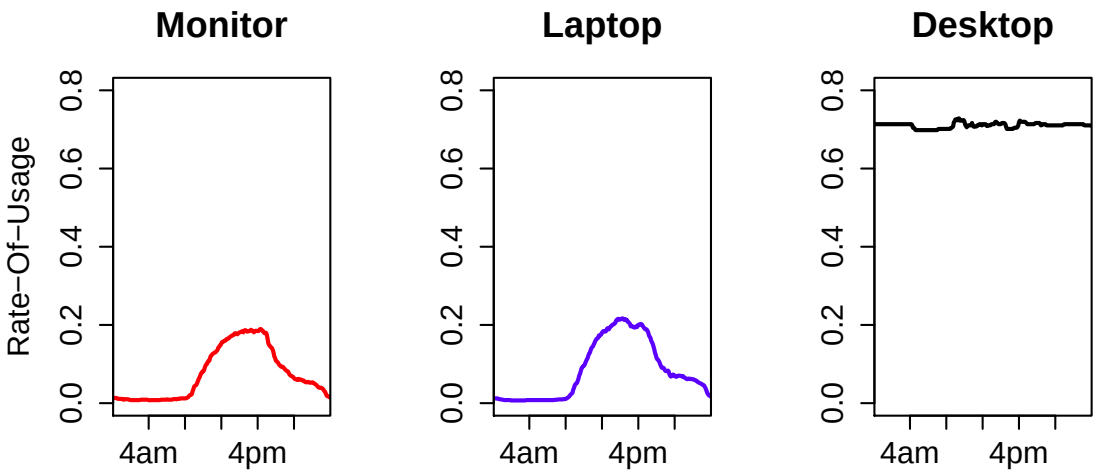

Figure 3.2: Rate-Of-Use of three types of appliances: monitor (left), laptop (middle) and desktop (right)

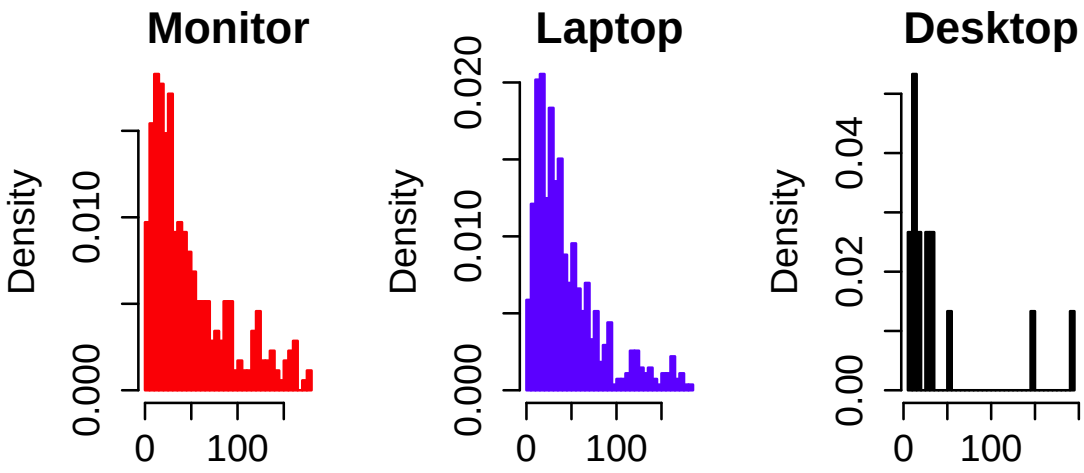

Figure 3.3: Histogram of duration statistics in minutes of three types of appliances: monitor (left), laptop (middle) and desktop (right). X axis is in 5 minutes interval

The results are shown in Figure 3.3 for office appliances. The limited capability to model the turn-off appliance events is a potential problem. Another issue of duration statistics is that they are usually time-dependent, which makes them costly to estimate.

ON/OFF-Probability Statistics

Another module focuses on the empirical ON/OFF-probability [62][75] (i.e. the probability of turning-ON/OFF at each time step).

Definition 3.3.2 (ON/OFF Probability)**.** *For certain appliance at t, the empirical ON/OFF probability is defined as $\widehat{P}_t^{\mathbf{ON/OFF}}$:*

$$\widehat{P}_t^{\mathbf{ON}} = \frac{\sum_{j=1}^m S_t^{(j)}(1-S_{t-1}^{(j)})}{\sum_{j=1}^m (1-S_{t-1}^{(j)})} = \frac{\overline{S_t} - \overline{S_t S_{t-1}}}{1-\overline{S_{t-1}}} \tag{3.2}$$

$$\widehat{P}_t^{\mathbf{OFF}} = \frac{\sum_{j=1}^m S_{t-1}^{(j)}(1-S_t^{(j)})}{\sum_{j=1}^m S_{t-1}^{(j)}} = \frac{\overline{S_{t-1}} - \overline{S_{t-1} S_t}}{\overline{S_{t-1}}} \tag{3.3}$$

with which we can do MC simulation to obtain the state sequences as a Markov Chain of all the appliances that we are interested in.

Definition 3.3.3 (Markov Chain)**.** *Markov Chain is a special case of a stochastic process. A stochastic process is a time sequence of variables $S_1, S_2, \cdots, S_t$, and their joint probability can be written as:*

$$\Pr(S_1, S_2, \cdots, S_t) = \Pr(S_1) \prod_{i=2}^{t} \Pr(S_i | S_{i-1}, \cdots, S_1)$$

A stochastic process is a Markov Chain (first order) if it follows the Markov property, in that $\Pr(S_i|S_{i-1}, \cdots, S_1) = \Pr(S_i|S_{i-1})$, and we have:

$$\Pr(S_1, S_2, \cdots, S_t) = \Pr(S_1) \prod_{i=2}^{t} \Pr(S_i | S_{i-1}) \tag{3.4}$$

Here $\Pr(S_i|S_{i-1})$ can also be viewed as transition probability. If they are consistent for all the i's, the Markov Chain is called Homogeneous Markov Chain; otherwise it is called Non-Homogeneous Markov Chain.

Definition 3.3.4. *After we run J MC simulations, we defined the simulated state in the j-th MC run as $\widehat{S}_{1:T}^j$, $j = 1, \cdots, J$.*

Compared to ROU model, the ON/OFF probability model can capture the usage pattern [39][62][75]. Previously, this model is built upon some time slots (e.g. "0∼8AM", "8∼9AM", "9∼11:30AM", "11:30∼1:30PM", "1:30∼5PM", "5∼7PM", "7∼9:30PM" and "9:30PM∼0AM"). The ON/OFF probability is assumed to be constant within each time slots.

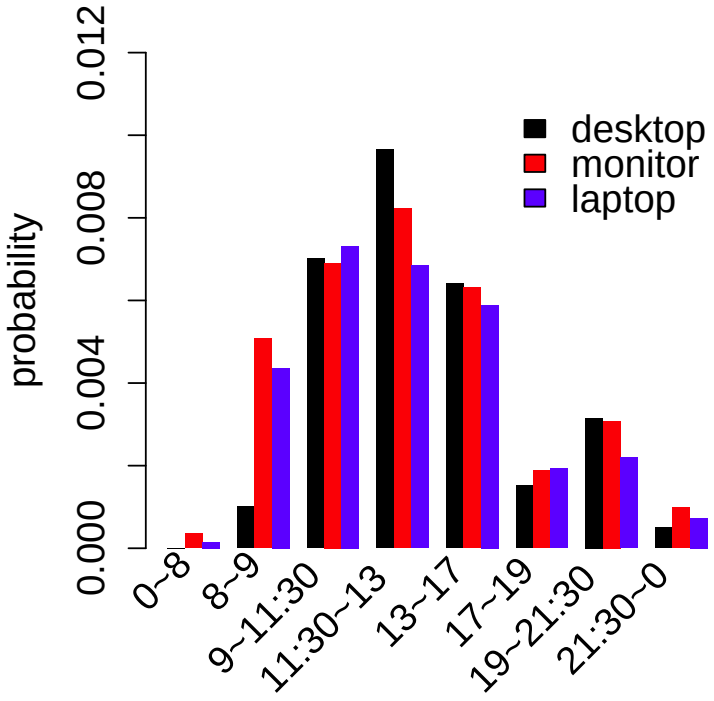

Figure 3.4: Time-dependent ON probability of three types of appliances: desktop (black), monitor (red) and laptop (blue)

The time-slot-based ON-probability $\widetilde{P}_t^{\mathbf{ON}}$ is shown in Figure 3.4, for desktop, monitor and laptop. Note that in Figure 3.2 the desktop pattern seems to be at constant line, which is due to the limited number of desktops in our test space, and because some of them are kept on overnight (i.e, their $\widetilde{P}_t^{\mathbf{OFF}}$ is small once they are ON). To simulate turning-ON, we use the probability of $\widetilde{P}_t^{\mathbf{ON}}/T_{\mathbf{SLOT}}$, in which $T_{\mathbf{SLOT}}$ is the length the time slots. For example, at time interval "8∼9AM", if we use 5 min interval step, $T_{\mathbf{SLOT}} = 12$.

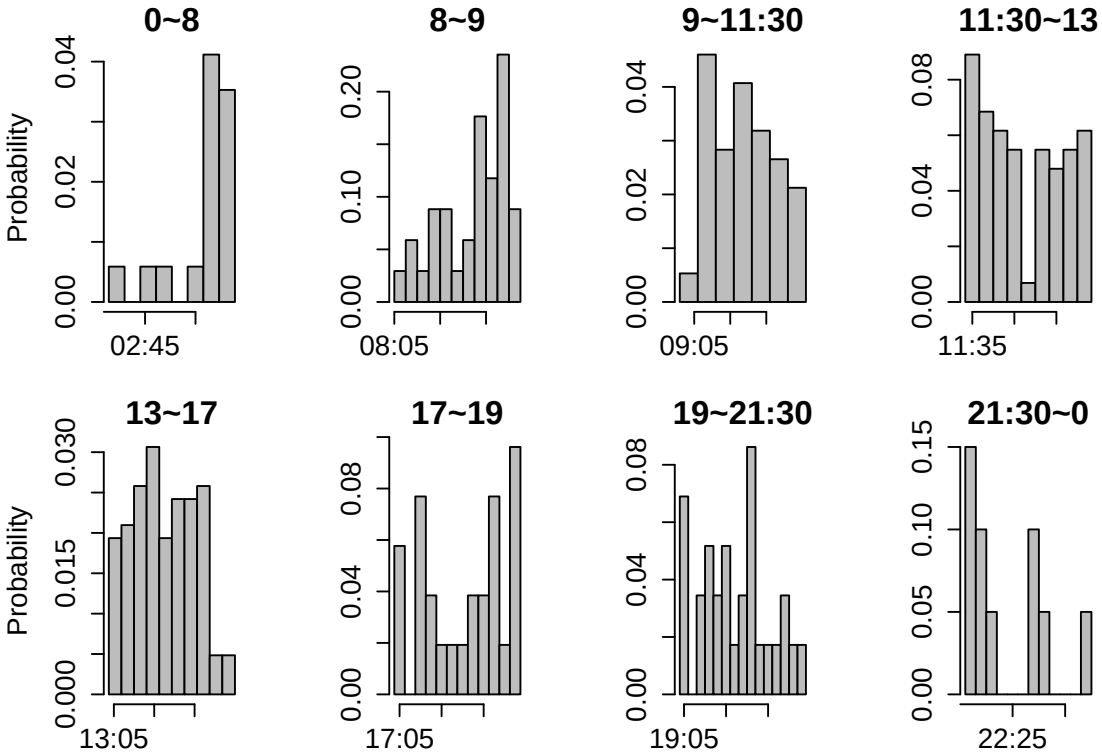

Figure 3.5: ON probability inside each time slot for monitor

One concern about the time-slot-based model is that the probability inside each slot is not captured well. According to a simple Poisson model, assuming independent events within each time slot, the ON events are geometrically distributed. However, as shown in Figure 3.5 where *monitor* is taken as an example, most events do not follow the model. The pattern of *laptop* and *desktop* can also demonstrate such discrepancy.

Appliance ON/OFF Probability Model

In our work, for statistical parameters, the appliance high-resolution ON/OFF probability model is used.

- On one hand, the ON/OFF states of the appliances are used, instead of the Time-Of-Use data in previous work. Hence, there is no empirical inference involved.

- On the other hand, the data collected in wireless sensor networks are used, which has resolution of up to one second per sample.

In our chosen model, we use an appliance-data-driven high-resolution ON/OFF probability model.

- We extract the probability that an appliance is present in some day, marked as $P_{\mathbf{PRES}}$, as well as the probability that an appliance is ON overnight, marked as $P_{\mathbf{INIT}}$. Then, from the wireless sensor network, we collect appliance power stream and build the model based on appliance information, instead of on activities (as presented in other works, in which an often-problematic activity-to-appliance transformation is needed [29]).
- âĂć Both ON/OFF probabilities are included and formulated in a Markov Chain framework, whereas duration statistics are not included. Therefore, we can better model the appliances' turning-OFF events.

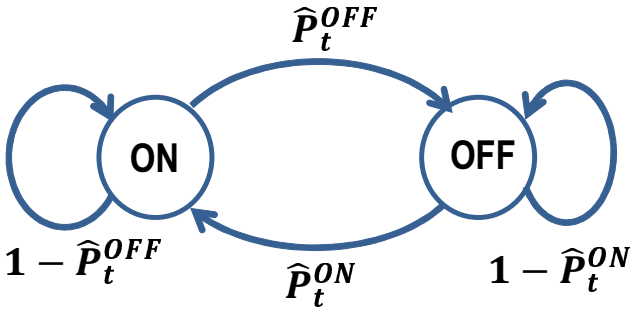

Figure 3.6: FSM interpretation of the model

- Instead of the time-slot model in Figure 3.4, we use a non-homogeneous Markov Chain model for both ON/OFF probabilities. For each appliance, the model can be interpreted as a two-state Finite State Machine (FSM) at each timestamp (Figure 3.6).

Power Estimation

Based on the FSM model, power consumption of a given space is estimated by running a Monte Carlo (MC) simulation to generate power sequences aggregated from individual appliances.

The MC-simulated appliance ON/OFF sequences (a) can capture non-homogeneous stochasticity of appliance usage patterns and is easily extended to analyze new techniques and policies, and (b) statistically converges to the ROU model in estimating states, which means this method is essentially reasonable in end-use energy profile modeling.

Theorem 3.3.1 (Convergence of MC Simulation)**.** *If $\widehat{S}^j_{1:n}$ is the j^{th} MC simulated time series from the FSM as in Figure 3.6 and we have J such MC simulations, then $\mathrm{E}[\frac{1}{J}\sum_j \widehat{S}^j_t] = \overline{S_t}$, in which $\overline{S_t}$ is the ROU, and $\lim_{J\to\infty}\mathrm{Var}(\frac{1}{J}\sum_j \widehat{S}^j_t) \to 0$. In other words, MC simulation converges a.s. to ROU.*

Proof. Let $\widehat{S}_1, \cdots, \widehat{S}_t$ be the states at different time steps from MC simulation. Assume that the states follows Markov Property, s.t. $\Pr(\widehat{S}_t|\widehat{S}_{t-1}, \cdots, \widehat{S}_1) = \Pr(\widehat{S}_t|\widehat{S}_{t-1})$. Then by the chain rule of expectation [63], we have:

$$\mathrm{E}[\widehat{S}_t] = \mathrm{E}[\mathrm{E}[\widehat{S}_t|\widehat{S}_{t-1}]] \tag{3.5}$$

Since we have:

$$\begin{aligned}\mathrm{E}[\widehat{S}_t|\widehat{S}_{t-1}] &= \Pr(\widehat{S}_t = 1|\widehat{S}_{t-1}) \\ &= \widehat{P}^{\mathbf{ON}}_t(1-\widehat{S}_{t-1}) + (1-\widehat{P}^{\mathbf{OFF}}_t)\widehat{S}_{t-1} \\ &= \widehat{P}^{\mathbf{ON}}_t + (1-\widehat{P}^{\mathbf{ON}}_t - \widehat{P}^{\mathbf{OFF}}_t)\widehat{S}_{t-1}\end{aligned} \tag{3.6}$$

Let us define G_t which follows as:

$$G_t = 1 - \widehat{P}^{\mathbf{ON}}_t - \widehat{P}^{\mathbf{OFF}}_t = \frac{\overline{S_t S_{t-1}} - \overline{S_t}\cdot\overline{S_{t-1}}}{(1-\overline{S_{t-1}})\overline{S_{t-1}}}$$

Then, combining (3.5) and (3.6) we obtain:

$$\mathrm{E}[\widehat{S}_t] = \widehat{P}^{\mathbf{ON}}_t + G_t\mathrm{E}[\widehat{S}_{t-1}] \tag{3.7}$$

Therefore, we can iteratively write $\mathrm{E}[\widehat{S}_t]$ as:

$$\mathbf{E}[\widehat{S}_t] = \widehat{P}^{\mathbf{ON}}_t + \sum_{\tau=3}^{t}\widehat{P}^{\mathbf{ON}}_{\tau-1}\prod_{i=\tau}^{t}G_i + \mathbf{E}[\widehat{S}_1]\prod_{i=2}^{t}G_i \tag{3.8}$$

The initial state at $t=1$ in MC simulation is generated from a Bernoulli process $p_1 = \mathbf{E}[\widehat{S}_1] = \overline{S_1}$. We put the expression of $\hat{P}^{\mathbf{ON/OFF}}_t$ as (3.2) and (3.3) in (3.8).

$$\widehat{P}^{\mathbf{ON}}_2\prod_{i=3}^{t}G_i + \overline{S_1}\prod_{i=2}^{t}G_i = \overline{S_2}\prod_{i=3}^{t}G_i \tag{3.9}$$

Then we have the following equation:

$$\mathbf{E}[\widehat{S}_t] = \widehat{P}_t^{\mathbf{ON}} + \sum_{\tau=4}^{t} \widehat{P}_{\tau-1}^{\mathbf{ON}} \prod_{i=\tau}^{t} G_i + \overline{S_2} \prod_{i=3}^{t} G_i$$

Therefore, we can simply equation (3.8) as:

$$\begin{aligned} \mathbf{E}[\widehat{S}_t] &= \widehat{P}_t^{\mathbf{ON}} + \overline{S_{t-1}} G_t \\ &= \frac{\overline{S_t} - \overline{S_t S_{t-1}}}{1 - \overline{S_{t-1}}} + \frac{\overline{S_t S_{t-1}} - \overline{S_t} \cdot \overline{S_{t-1}}}{1 - \overline{S_{t-1}}} = \overline{S_t} \end{aligned} \tag{3.10}$$

Since $\widehat{S}_t^j$s are all binary sequences, $\mathrm{Var}(\widehat{S}_t^j) = \overline{S_t}(1 - \overline{S_t})$ and naturally we have

$$\lim_{J\to\infty} \mathrm{Var}(\frac{1}{J} \sum_j \widehat{S}_t^j) = \lim_{J\to\infty} \frac{1}{J} \mathrm{Var}(\widehat{S}_t^j) \to 0 \tag{3.11}$$

Thus, MC simulation converges to the ROU. It should, however, be noted that Theorem 3.3.1 holds only if the ON/OFF probabilities are consistent between simulation and observation. □

Data Sparsity & Kernel Smoothing

The ON/OFF events are always sparse [41], and variance of the estimation is always high. In this situation, smoothing is needed.

When there are large amount of spikes, the empirical probability function can be smoothed by a Kernel Smoother to obtain the probability function.

$$\widetilde{P}_t^{\mathbf{ON/OFF}} = \frac{\sum_{i=1}^{n} K(t,i) \widehat{P}_i^{\mathbf{ON/OFF}}}{\sum_{i=1}^{n} K(t,i)} \tag{3.12}$$

in which $K(t,i)$ is the kernel function. Usually we use Gaussian kernel $K(t,i) = \exp\left(-\frac{(i-t)^2}{2h^2}\right)$ in which h is the bandwidth. The larger the bandwidth, the more smoothing the kernel does. h can be chosen as the plug-in bandwidth (hpi) [71].

Remark 3.3.1. *If we use $\widetilde{P}_i^{\mathbf{ON/OFF}}$ instead of $\widehat{P}_i^{\mathbf{ON/OFF}}$, Theorem 3.3.1 no longer holds. However, under some most basic regularity condition of the function $\widehat{P}_i^{\mathbf{ON/OFF}}$, we have the following relationship:*

$$\lim_{h\to 0} \widetilde{P}_i^{\mathbf{ON/OFF}} \to \widehat{P}_i^{\mathbf{ON/OFF}}$$

This means, under reasonably chosen bandwidth of the function $K(\cdot)$, the smoothed probabilities in (3.13) *will be reasonable approximation for $\widehat{P}_i^{\mathbf{ON/OFF}}$, and Theorem 3.3.1 will also* approximately *holds. It should be noted here that a strict analysis on the condition of the bandwidth would be required to fully understand the performance of smoothing, and because of the scope of this work, this will be a subject of future work.*

Modeling of Cross-Correlation

In this study's experimental space, especially for computer-related appliances, we have 11 monitors, 5 desktops, and 14 laptops[1]. Intuitively, we can simulate each appliance independently and aggregate them to get the full power consumption value. The mean of the aggregation, as a corollary of Theorem 3.3.1, is unbiased. The variance, however, could be underestimated. Cross-correlation among appliances needs to be addressed. Here in this study, there are two reasonable assumptions.

- The appliances in the same category (monitors, desktops, or laptops) are the same type[2]
- The correlation pattern is homogeneous, which means it is same for every day.

An intuitive way to analyze this information is to generate correlated Bernoulli sequences in Monte Carlo simulation [50]. However, for multivariate non-homogeneous Markov Chain, generation such correlated Bernoulli sequences is difficult and unreliable [50]. In this work, we propose a way to correct the variance on the independently simulated sequences.

For example, let $S_{t,i}$ be the state of i-th single appliance, its variance $\mathrm{Var}\left(S_{t,i}\right) = \sigma_g^2$ we already know, $g \in \{desktop, monitor, laptop\}$ is the appliance type, then the aggregated variance of p different appliances is:

$$\mathrm{Var}\left(\sum_{i=1}^{p} S_{t,i}\right) = \sum_{i=1}^{p} \sigma_{t,a(i)}^2 + \sum_{i \neq j} \mathrm{cov}\left(S_{t,i}, S_{t,j}\right) \tag{3.13}$$

in which $a(i)$ is the type of the i-th appliance, The second term on RHS corresponds to the covariance between different appliances, and should be added to avoid underestimation of overall variation. This term can be extracted directly from historical data.

[1] The cross-correlation among lighting and shared appliances are not of significance

[2] This is reasonable especially for office buildings when occupants have roughly the same sets of appliances.

Results and Discussion

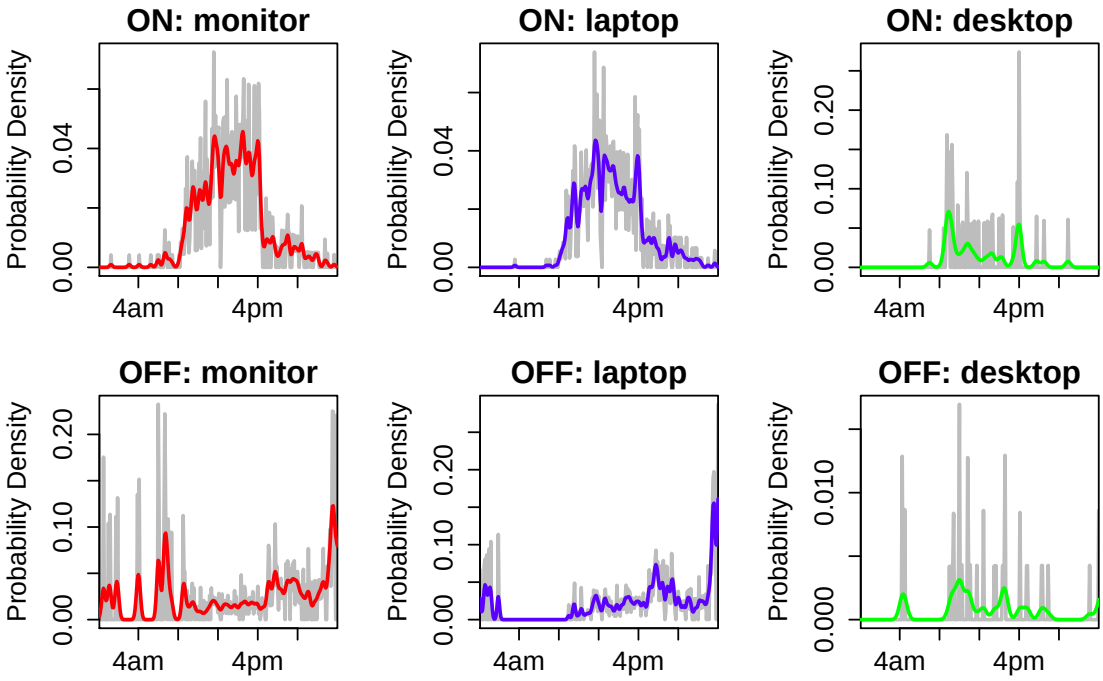

Figure 3.7: ON/OFF Probability in 5 min interval for Monitor, Laptop, and Desktop. Gray lines: Measurement; Colored lines: Kernel smoothed

- Office Appliances: Office Appliances: The office appliances include *monitor, laptop, and desktop*. The estimated ON/OFF probabilities for these three types of appliances are shown in Figure 3.7. It is observed that the ON probability peaks in the early morning and decreases during the day, whereas the OFF probability peaks later in the day. It should be noted that data regarding desktop is sparse and ON/OFF probabilities contain more uncertainty. Only weekdays are included in this study.

- Pathway/Room Lighting: Pathway/Room Lighting: Lighting power consumption is a major contributor to a building's energy profile. In this study's test space in Cory 406 at UC Berkeley, there is pathway lighting and room lighting. Pathway lighting is shared in a large working area and has a more standard schedule throughout the day. Room lighting has a motion sensor, so it is more

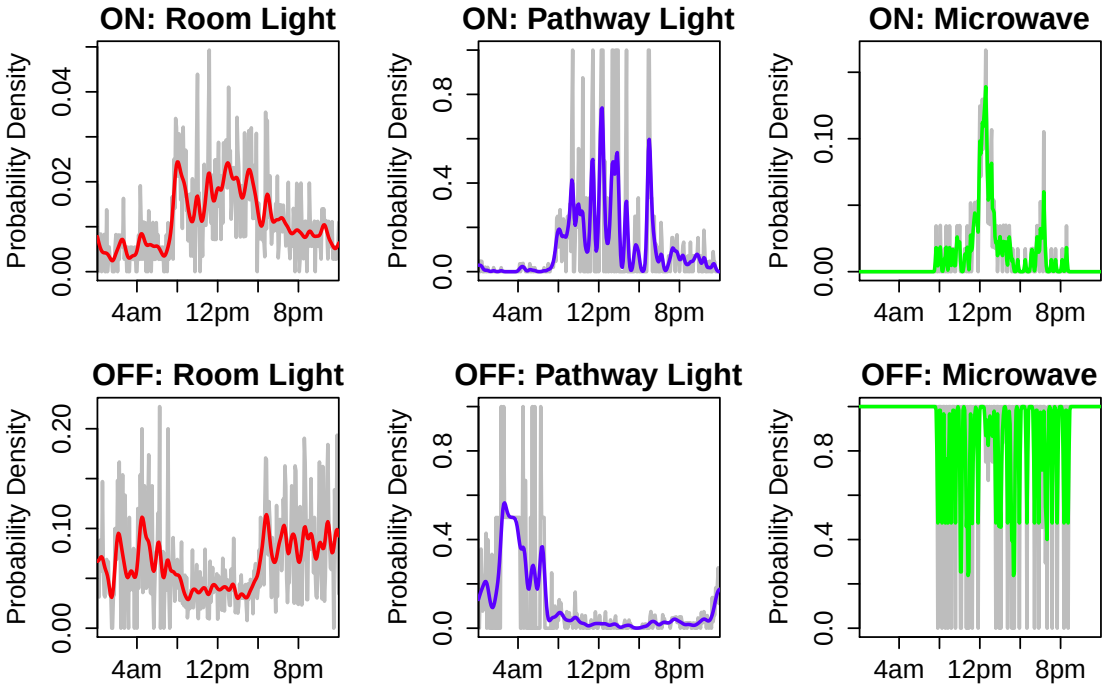

Figure 3.8: ON/OFF Probability in 5 min interval for Room lighting, Pathway lighting and Microwave. Gray lines: Measurement; Colored lines: Kernel smoothed

adaptive to occupant behavior. The PowerScout data we collected contains the aggregated signal of lighting power in seven rooms. For model simplicity, we will assume that the seven rooms are the same. The result is shown in Figure 3.8. The pathway lighting has little overnight activity, and the estimation has more bias, since in (3.3), $\overline{S_t}$ is zero for some t. These data points are given a probability of 0.5.

- Shared Appliances: Shared appliances include a microwave, a water heater, a coffee maker, and a refrigerator. The water heater and refrigerator have a strong periodic pattern and less dependency on occupant behavior. The microwave and coffee maker show a spike-like pattern. The estimated probability density for a microwave is shown in Figure 3.8. Notice that the OFF probability is very high, since the duration of each ON event is usually very short, as compared to our five-minute estimation interval.

  Actually, for those appliances, duration is roughly fixed depending on the ap-

pliance settings. In the next Section, we will discuss an alternative way to model the appliances, based on a non-homogeneous Poisson Process model.

It is expected that in a larger office building, when more appliances are present, our proposed model would be more capable to capture overnight patterns. Moreover, it should be noted that when the building occupancy schematic changes, the only thing that needs to be tuned is the building profile. As long as we have a reasonable category of users, we can evaluate the building energy performance accordingly.

3.4 Shared Appliances

Poisson Process Model

As mentioned in Section 3.5, shared appliances (e.g., microwaves, printers, coffee machines) usually demonstrate spiking patterns. The duration of the spike is usually due to a machine's setting. The bottleneck of the modeling is, instead, the turning-ON probability of the appliance. Since several people are sharing this appliance, we would like to filter out an individual turning-ON probability or other usage characteristic that is independent of the number of users. That way, we are able to extend this model to another building space. This work shows a methodology to model the shared appliances and eventually filter out a usage pattern of a shared appliance from *one* occupant.

Essentially, we model the usage of an appliance through a Poisson process [63], with the rate of the process depending on the number of occupants inside a space. The Poisson Process (PP) models the number of events n_t during a $[0, t]$ interval, and n_t follows the Poisson distribution with rate λt as $n_t \sim \text{Pois}(\lambda t)$ as:

$$\Pr(n_t = k) = \frac{\lambda^k t^k}{k!} e^{-\lambda t}$$

in which λ is the rate function. The expectation of the number of events is λt, and the variance is λt as well. PP is a memoryless process [63], which means in the time interval $[s, t + s]$, the incremental events satisfies Poisson distribution $\text{Pois}(\lambda t)$ as well:

$$\Pr(n_{t+s} - n_s = k) = \frac{\lambda^k t^k}{k!} e^{-\lambda t} \tag{3.14}$$

If we model each user as a PP, and we have l identical users in total, the aggregation is still PP as $\sum_{k=1}^{l} n_t^{(k)} \sim \text{Pois}(l\lambda t)$.

Non-Homogeneous Poisson Process (NHPP) Model

If the rate function is time dependent $\lambda(t)$, then the process is Non-Homogeneous Poisson Process (NHPP). For instance, given the rate function at time t as $\lambda(t)$, the number of events in a small interval $[t, t+h]$ follows $n_{t+h} - n_t \sim \text{Pois}(\lambda(t)h)$. If we argue that h is one unit of time, then the expectation would be $\mathbf{E}[n_{t+h} - n_t] = \lambda(t)$. Hence, if we assume the number of occupants is also time dependent function $\Theta(t)$, then $n_{t+h} - n_t \sim \text{Pois}(\lambda(t)\Theta(t)h)$. From this stand point, we only need to estimate the time-dependent rate function $\lambda(t)$ in order to extract an *individual* usage pattern.

Based on this model, we construct a relationship between the events and the time dependent rate function $\lambda(t)$. If we would like to estimation the time-dependent rate function $\lambda(t)$, we can obtain it through statistical inference.

Bayesian Statistics framework

Based on NHPP model, we can construct the full probability function in Bayesian framework. Let $v(t)$ be the incremental number of events at time t, and let the joint prior probability function for $\lambda(t)$ and $\theta(t)$ as $\phi(\lambda(t), \theta(t))$, we can write the full probability function of $n(t)$, $\lambda(t)$ and $\theta(t)$ as:

$$\Pr \propto \prod_{t=1}^{n} e^{-\lambda(t)\theta(t)} [\lambda(t)\theta(t)]^{v(t)} \phi(\lambda(t), \theta(t)) \tag{3.15}$$

If we further assumes that $\theta(t)$ is the daily number of occupants, which means it is constant within a day; whereas $\lambda(t)$ is time-of-day dependent and same for each day, and let $\lambda(t) = \lambda_0 \eta(t)$ for simplicity, in which $\eta(t) \in \{\eta_1, \cdots, \eta_p\}$ is a normalized data with $\sum_j \eta_j = p$ the daily data points[3], and $\theta(t) \in \{\theta_1, \cdots, \theta_d\}$ as d the total number of days. Let v be total number of events, v_j corresponds to the number of events w.r.t. the j-th time slots (such that $\sum_j v_j = v$), and $v^{(i)}$ be the number of events on the i-th day, we have:

$$\Pr \propto e^{-\lambda_0 p \sum_i \theta_i} \lambda_0^v \left(\prod_j \eta_j^{v_j} \right) \left(\prod_i \theta_i^{v^{(i)}} \right) \phi(\cdot) \tag{3.16}$$

We can use MCMC techniques such as Gibbs sampler to generate sample of $\lambda_0, \{\eta_j\}_{j=1}^p$ and $\{\theta_i\}_{j=i}^d$. The prior of Poisson distribution is the Gamma distribution [63], we follow the process as below:

[3] In 5 min interval data, there are 288 date point every day, so $P = 288$.

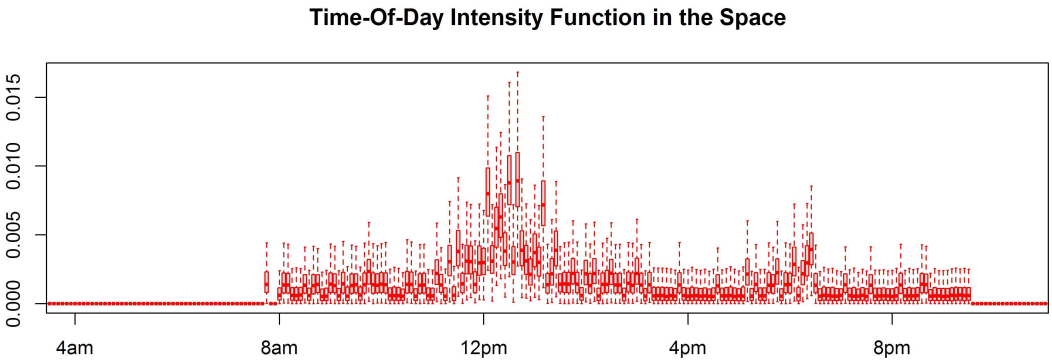

Figure 3.9: Sampling result of $\lambda(t)$ along the day with 5 min per sample.

- sample $\lambda_0 \sim \Gamma(\alpha^\lambda + v, \beta^\lambda + p\sum_i \theta_i)$, which assumes a prior of $\Gamma(\alpha^\lambda, \beta^\lambda)$
- sample $\eta_1, \cdots, \eta_p$, in which $\eta_j \sim \text{Dir}(\alpha^\eta + v_j)$ which is the Dirichlet distribution with prior $\text{Dir}(\alpha^\eta)$.
- sample $\theta_1, \cdots, \theta_d$ in which $\theta_i \sim \Gamma(\alpha^{\theta_i} + v^{(i)}, \beta^{\theta_i} + p\lambda_0)$, which assumes a prior of $\Gamma(\alpha^{\theta_k}, \beta^{\theta_k})$.

Results and Discussion

Prior distributions are assumed from rough understanding, and we run 1500 MCMC steps, with another 500 as burn-in period.

The sampling result of $\lambda(t)$ $(t = 1, \cdot, p)$ is shown in Figure 3.9. The histogram of λ_0 in Equation (3.16) is shown in Figure 3.10 for reference. We can tell the daily pattern from Figure 3.9 as well as the daily fluctuation in distribution.

The sampling result of $\theta(t)$ (actually $\theta_1, \cdots, \theta_d$) is shown in Figure 3.11. We can also obtain the statistical variability from the figure. This indicates that the sampling method not only can give estimation of the parameters, but also give estimation of their statistical variability.

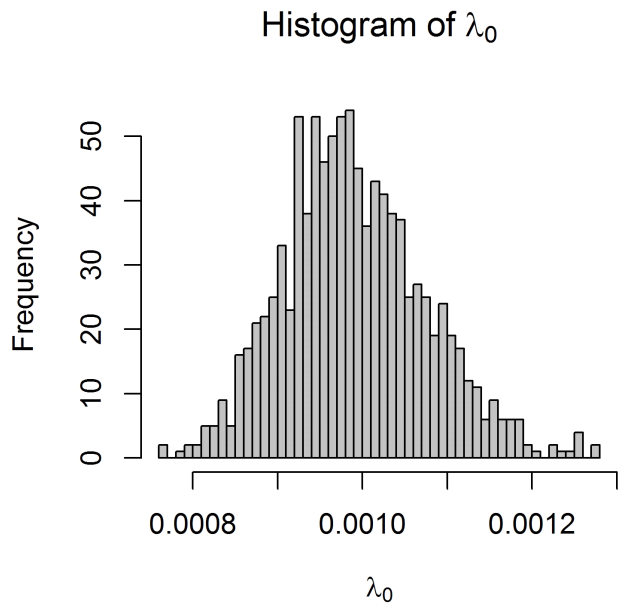

Figure 3.10: Sampling histogram of λ_0.

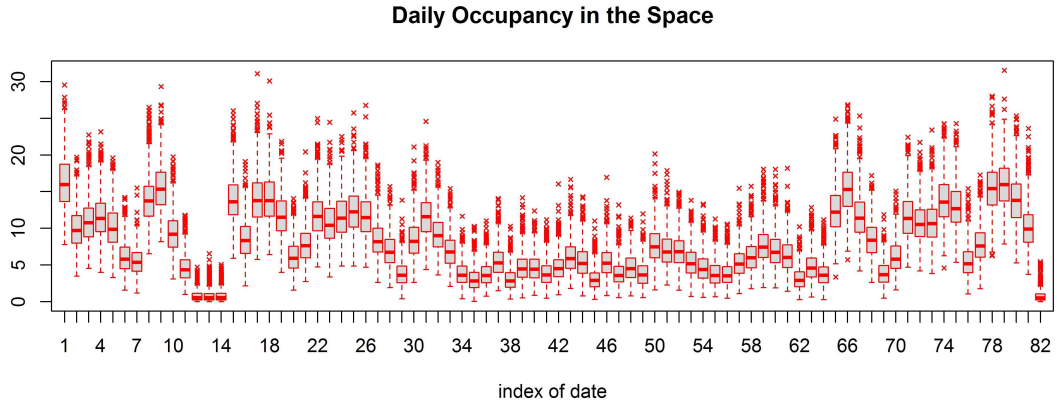

Figure 3.11: Sampling result of $\Theta(t)$ in each day.

3.5 Conclusion

In this chapter, *modeling* issues under the *Bottom-Up* setting are comprehensively discussed. Compared to the Time-Of-Use (TOU) data used in previous *Bottom-Up* models, this work takes advantages of the high frequency sampled data from wireless sensor networks and builds an appliance-data-driven end-use model. ON/OFF probabilities of appliances are extracted, and a theoretically unbiased Finite-State-Machine (FSM) Monte Carlo model is developed with cross-correlation correction. This chapter also briefly introduces work on modeling the shared appliance based on the Non-Homogeneous Poisson Process (NHPP) sampling method, which can filter out an individual usage pattern out of ON/OFF states of shared appliances.

Chapter 4

Bottom-Up End-Use Model: Data and Experiments

In Chapter 3, we studied the theories and settings of the *Bottom-Up* end-use model and demonstrated that the *Bottom-Up* model can be re-used in other similar buildings to estimate end-use power consumption. This section continues this line of study by pulling everything together and verifies model performance in a real end-use modeling application.

In this work, we make use of the Bottom-Up model structure shown in Chapter 3, and only *Level-II* and *Level-III* models are focused on here. For convenience, the proposed model's schematic is shown again in Fig. 4.1.

In Section 4.1, the data collection process is discussed. Then in Section 4.2, a brief discussion about the power disaggregation technique used to filter out individual appliance ON/OFF states from aggregated raw power sequences follows. Finally, in section 4.3, experimental results are shown as happened in Cory Hall and Sutardja-Dai Hall at UC Berkeley, followed by the conclusion in Section 4.3.

4.1 Data Collection

Power consumption of the appliances is collected through a large-scale wireless sensor network (WSN). WSNs have been implemented in many different scenarios to facilitate system estimation, conditioning, and diagnosis [39][45][38].

– DENT meter [17] is used to collect whole space real-time power consumption data. The DENT meter has 18 channels, each one monitoring a

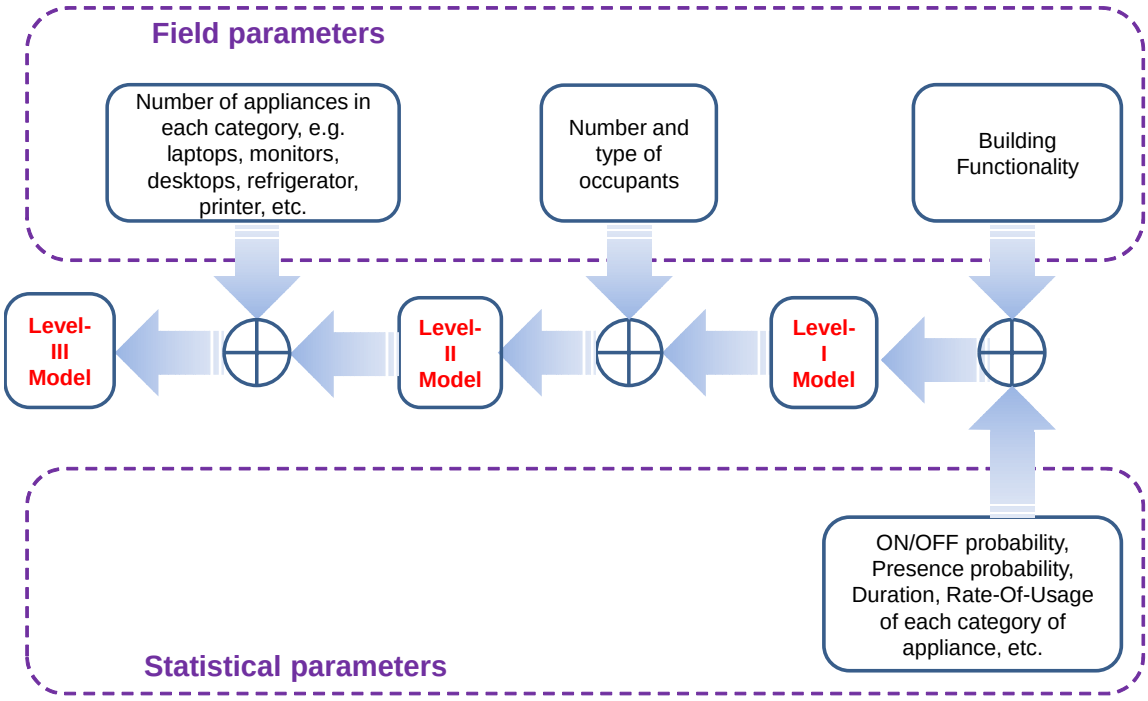

Figure 4.1: Parameters in *Bottom-up* model: Field Parameters and Statistical Parameters. *Level-III* model is the most complex, and *Level-II* model is less complex; *Level-I* is the simplest but rather hard to achieve.

subset of appliances, e.g. plug loads, lights, kitchenware etc. The DENT meter data is handled in CoreSight from OSIsoft[1].

– ACme sensors are used to collect real-time power consumption of each occupant [33], with resolution up to one second per sample. The ACme meter data is handled using the sMAP protocol [16]. We implement one ACme sensor for each occupant to optimize cost and experimental performance. The states of each appliance are filtered out by the power disaggregation algorithm from the aggregated occupant-level power consumption, as will be illustrated in the next section [37].

[1] http://picoresight.osisoft.com/

4.2 Power Disaggregation

In *Bottom-Up* models, we need to collect power consumption data of each individual appliance. However, deploying sensors to each appliance will be costly and raise issue in privacy and stability, especially in modern commercial buildings.

On one hand, modern buildings demonstrate sophisticated functionality and increasing number of appliances, which makes large-scale sensor deployment costly. On the other hand, users may complain if many sensors are deployed in their space. Finally yet importantly, as more sensors are included in the network, communication may suffer from stability issues, and thus data quality is less guaranteed [45].

For all of these reasons above, a low-cost, non-intrusive monitoring is preferred that can measure power consumption of appliances without the direct attachment of power meters [76]. The most common solution to this issue is to use a power strip to aggregate all the appliances of each user and attach it to a power meter to measure aggregated power. Then, we apply *power disaggregation* methods to the aggregated signal to obtain signals of each individual appliance, as shown in Figure 4.2. With the appliance-level consumption recovered, we can build a *Bottom-up* model for the building space under study. Here is a mathematical definition of power disaggregation:

Definition 4.2.1 (Power Disaggregation)**.** *In power disaggregation, we* decode *the ON/OFF state of individual appliance from an observed* aggregated *power stream. Let* $p_t, \forall t = 1, \cdots, n$ *be the aggregated power stream from* p *appliances. Let* $\mathbf{S}_t$ *be the state vector of the* n *appliances at step* t*. Our task is to infer* $\mathbf{S}_t$ *from* p_t*.* $\mathbf{S}_t$ *is a vector of* n *binary variables, one for each appliance, i.e.* $\mathbf{S}_t \in \{0, 1\}^p$*, in which* 1 *for ON,* 0 *for OFF. There are in total* 2^p *combinations of ON/OFF states.*

Various existing power disaggregation methods are studied in this section, along with a comparison of their performance followed by proposed new algorithms based on sequential hypothesis testing.

Related Work

Typical solutions to Power Disaggregation are either based on a *Hidden Markov Model*, or on *Edge-based Model*.

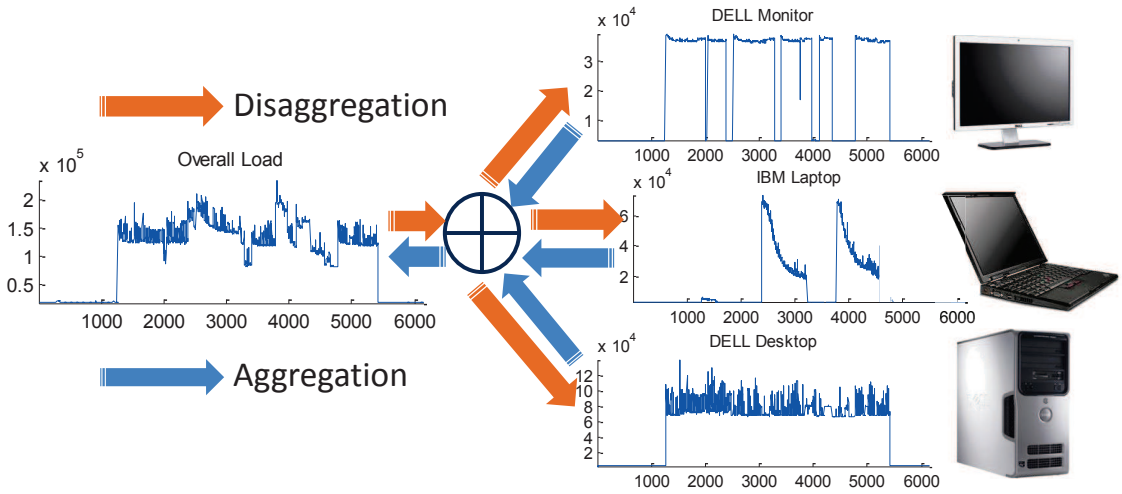

Figure 4.2: Schematics of power disaggregation, decoding aggregated power stream (including a desktop, a monitor and a laptop) to appliance-level streams

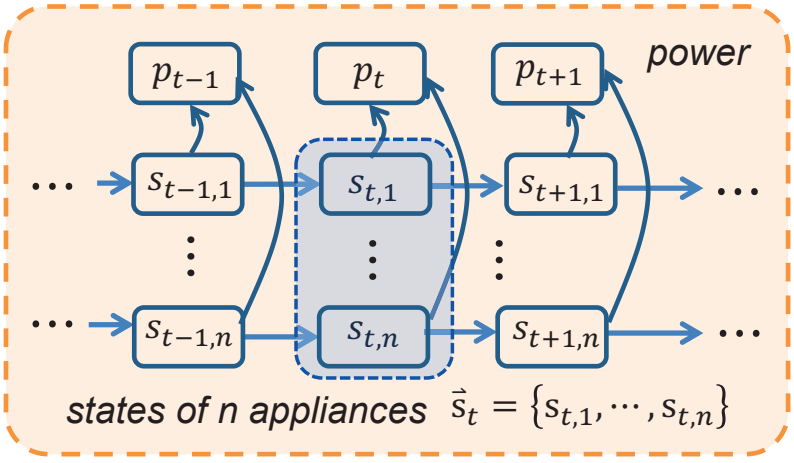

Figure 4.3: Schematics of Hidden Markov Model for power consumption over time $(p_t, t = 1, \cdots, T)$

– Hidden Markov Model (HMM): The aggregated power stream is modeled as a Hidden Markov Chain (HMC), with hidden states as the ON/OFF states of individual appliances, as shown in Figure 4.3 [25].

  Firstly, the aggregated power p_t is a Gaussian distributed variable conditioned on the appliance state vector $\mathbf{s}_t$. If we assume that the power

consumption of the i-th appliance is approximately Gaussian distributed as $\mathcal{N}(W_i, \sigma_i^2)$, and let $\mathbf{w} = \{W_1, \cdots, W_n\}$ and $\mathbf{\Sigma} = \{\sigma_1^2, \cdots, \sigma_n^2\}$, then the aggregated power follows:

$$p_t|s \sim \mathcal{N}(\mathbf{s}^T\mathbf{w}, \mathbf{s}^T\mathbf{\Sigma}) \tag{4.1}$$

Secondly, the sequence of $\mathbf{s}_t$ with $t = 1, \cdots, T$ in Figure is modeled as a Markov Chain (MC).

Definition 4.2.2 (Markov Chain)**.** *Markov Chain is a special case of a stochastic process. A stochastic process is a time sequence of variables $S_1, S_2, \cdots, S_t$, and their joint probability can be written as:*

$$\Pr(S_1, S_2, \cdots, S_t) = \Pr(S_1) \prod_{i=2}^{t} \Pr(S_i|S_{i-1}, \cdots, S_1)$$

A stochastic process is a Markov Chain (first order) if it follows the Markov property, in that $\Pr(S_i|S_{i-1}, \cdots, S_1) = \Pr(S_i|S_{i-1})$, and we have:

$$\Pr(S_1, S_2, \cdots, S_t) = \Pr(S_1) \prod_{i=2}^{t} \Pr(S_i|S_{i-1})$$

Here $\Pr(S_i|S_{i-1})$ can also be viewed as transition probability. If they are consistent for all the i's, the Markov Chain is called Homogeneous Markov Chain; otherwise it is called Non-Homogeneous Markov Chain.

For convenience, we note that:

$$\Pr(\mathbf{s}_t|\mathbf{s}_{t-1}, \cdots, \mathbf{s}_1) = \Pi_{\mathbf{s}_{t-1}, \mathbf{s}_t} \tag{4.2}$$

Based on (4.1) and (4.2), we estimate the state at each step, based on Maximum Likelihood Estimation (MLE) estimation of $\mathbf{s}_t$:

$$\mathbf{s}_t = arg \max_{\mathbf{s}} \Pr(\mathbf{s}_t = \mathbf{s}|p_{1:T}) \tag{4.3}$$

Since the search space is 2^n, there will be an exponential explosion w.r.t. n. However, if we assume that only one appliance is switching at each step, the incremental state search space from $\mathbf{s}_{t-1}$ to $\mathbf{s}_t$ is only n. This assumption is reasonable with manually switched devices and a sampling rate at the sensor node higher than 1 sec/sample.

Equation (4.3) can be solved by a Viterbi algorithm [25].

Definition 4.2.3 (Viterbi Algorithms). *We note* $L_t(\mathbf{s}) = \Pr(\mathbf{s}_t = \mathbf{s}|\mathbf{p}_{1:T})$ *as the likelihood function and we use* $\mathbf{s}_{t-1,\boldsymbol{ML}} = \Psi_t(\mathbf{s}_t)$ *to store the most likely state back at step* $t-1$ *given that the current state at* t *is* $\mathbf{s}_t$*. Then, it is argued that* $L_t(\mathbf{s})$ *and* $\Psi_t(\mathbf{s}_t)$ *can be obtained from the terms from step* $t-1$*:*

$$\begin{cases} L_t(\mathbf{s}) & = \max_{\mathbf{s}'} L_{t-1}(\mathbf{s}')\Pi_{\mathbf{s}',\mathbf{s}} \Pr(p_t|\mathbf{s}_t = \mathbf{s}) \\ \Psi_t(\mathbf{s}_t) & = arg\max_{\mathbf{s}'} L_{t-1}(\mathbf{s}')\Pi_{\mathbf{s}',\mathbf{s}_t} \end{cases} \tag{4.4}$$

The above problem is solved sequentially, as first estimate the state at the last step $\mathbf{s}_T = arg\max_{\mathbf{s}'} L_T(\mathbf{s}')$*, and then backtrack for the best estimate at each step as* $\mathbf{s}_{t-1} = \Psi_t(\mathbf{s}_t)$*.*

HMM gives stable state inference, and many existing algorithms on power disaggregation are built upon this basic model. Wang *et al.* [72] treated power disaggregation in a convex optimization framework using sparse constraints. [59] solved the HMM by the Extended Viterbi algorithm and considered only the major power consuming appliances. The sampling method is widely used to deal with the exponential explosion issue. In [41][44][43][34], statistical inference of the joint distribution is based on Factorial HMM [28], though most of the sampling methods have computation issues.

However, the standard HMM does not have a good way to handle the fact that states may stay unchanged for long time intervals. This is significant for our problem, since many appliances, such as a lamp or a monitor, will have very different duration characteristics, while HMM models the duration as a Geometric distribution [44]. Some extensions of HMM have been proposed to address this issue. In [27], the persistence of state (stickiness) is guaranteed by introducing a constraint on the Markov chain model. Whereas in [44], [43], a Hidden Semi-Markov Model is used to model duration statistics. However, in most cases, we need a long training period of time of this model, since ON/OFF events of individual appliances may not be that frequent.

– Edge-based Model: An intuitive way to get around duration modeling is to focus only on the ON/OFF edges, in an approach we call the Edge-based model, as shown in Figure 4.4. Edge-based model applies a change detection algorithm to track the edges and trace the source based on statistical learning methods [4]. Usually, we track the mean (β_t) and variance (σ_t^2) of the aggregated power over time using an exponential

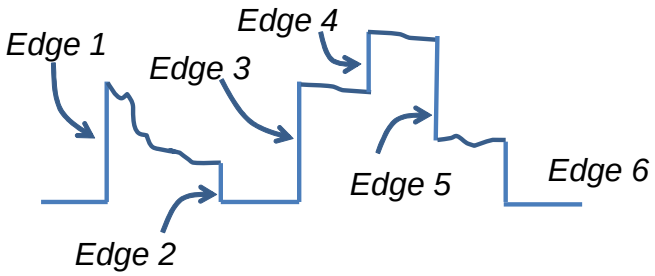

Figure 4.4: Schematics of Edge-based method

moving average filter such as:

$$\begin{cases} \beta_0 &= \frac{1}{d}\sum_{\tau=t-1}^{t-d} p_\tau \exp\left(-\frac{\tau-t}{\omega}\right) \\ \sigma_0^2 &= \frac{1}{d}\sum_{\tau=t-1}^{t-d} (p_\tau - \beta_0)^2 \exp\left(-\frac{\tau-t}{\omega}\right) \end{cases} \tag{4.5}$$

where ω is the decay factor and d the window size. Then, we look at the deviation of the current power p_t w.r.t. the mean and variance [7]. Edge-based model originates from the early work on NILM [30]. A review can be found in [76]. Algorithms that are studied include Linear Discriminant Classifier [20], Bayes classifier [22], Neural Network [21], etc.

Around the edges, there are several transient features that can be extracted from the active power or the reactive power readings, the latter often having unique harmonic patterns when observed at high enough sampling rates [48]. Such high frequency transients can help distinguish between, for example of a coffee-maker and a chandelier, especially when focusing on their reactive power patterns.

In general, high frequency sampling will also be useful in distinguishing between appliances, since larger data sets, aided by the *Law of Large Numbers* [7], will generally be better for distinguishing among different sources. The obvious tradeoff here is, of course, that higher sampling rates would typically imply higher instrumentation and computational cost.

Existing Challenges

In general, the existing challenges are from the noise and the non-stationarity.

– We have the assumption that the power consumption is essentially a Gaussian random variable. However, in a real power system, the noise is an

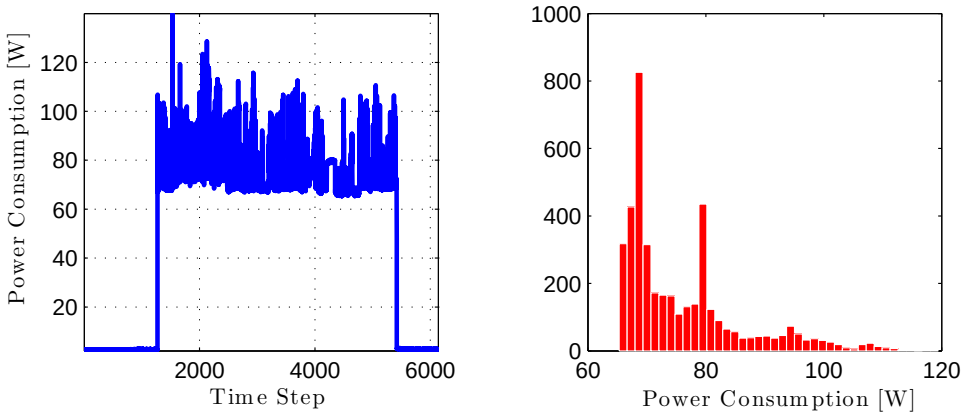

Figure 4.5: Impulse noise observed in power consumption data

approximate Gaussian noise plus large amounts of spikes or impulses. In Figure 4.5, we can see that the spikes apparently deviate from Gaussian distribution and can be as high as 5∼10 Watts. This deviation from Gaussian noise would cause unexpected trouble to the performance of a power disaggregation algorithm.

– Another assumption is that the power consumption is stationary. An appliance with multiple levels of power consumption curve can be modeled as a Gaussian mixture. However, many power consumption curves follow temporal trends or fluctuations, as shown in Figure 4.6. We call this phenomenon non-stationarity. This would make the traditional power disaggregation fail.

In this work, we will be focusing on addressing those issues. A robust sequential test-based method will be proposed.

Sequential Test Based Power Disaggregation: Theory

From the statistics perspective, edge detection is inherently a hypothesis testing problem [7]. The null hypothesis is *no change happened* (H_0), and the alternative hypothesis is *change happened* (H_1). Hypothesis testing for change detection has been widely studied before [4][7]. Usually we design a test statistic $T(x)$. If and only if $T(x) > \lambda$, H_0 is rejected; whereas if $T(x) \leq \lambda$, we still keep H_0, in which λ is the threshold. To evaluate the test, we use the *power* of the test and the False Positive Rate (FPR).

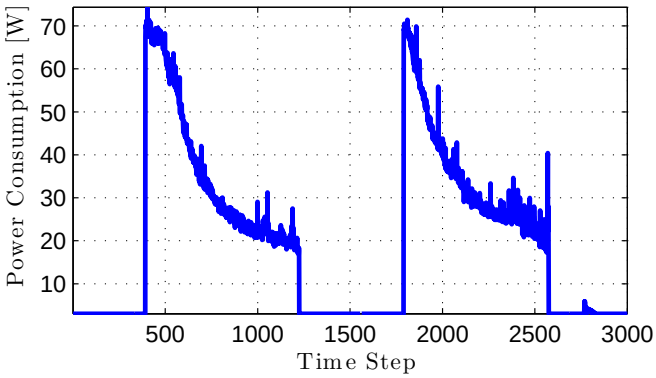

Figure 4.6: Non-stationarity observed in power consumption data, in both periodicity, trending, and chaotic way

Definition 4.2.4 (Power of Test). *The* power *of a test (β) is defined as the probability that it will* correctly *reject the null hypothesis. Mathematically, it is formed as:*

$$\beta = \Pr(T(x) > c|H_1) \tag{4.6}$$

Definition 4.2.5 (False Positive Rate). *Moreover, we define the False Positive Rate (FPR), which determines the error rate of a hypothesis test as:*

$$\alpha = \Pr(T(x) > c|H_0) \tag{4.7}$$

The test statistic $T(x)$ determines the power and error rate of the test. In [7], it has been argued that within all the test statistics, Neyman-Pearson framework is the most powerful.

Definition 4.2.6 (Neyman-Pearson Test). *Let the probability density and parameter be $f_0(x)$ and θ_0 for H_0, respectively, and $f_1(x)$ and θ_1 for H_1, respectively. The N-P framework ensures that the Uniformly Most Powerful (UMP) test given certain False Positive Rate (FPR) is achieved by using Probability Ratio as test statistic, i.e. $T(x) = \frac{f_1(x)}{f_0(x)}$. The result of the test (noted as a 0/1 variable $\delta(x)$) follows:*

$$\delta(x) = \begin{cases} 1 & \text{if } T(x) > \lambda \text{ i.e. reject } H_0 \\ 0 & \text{if } T(x) < \lambda \text{ i.e. do not reject } H_0 \end{cases} \tag{4.8}$$

where the value of λ is determined from the constraint of FPR $\alpha = \Pr(T(x) > \lambda|H_0)$. However, the power of the N-P test depends on the sample size of the input data x, which limits the performance of the test.

Definition 4.2.7 (Sequential N-P Test)**.** *The sample size issue can be solved by the sequential version of the N-P Test, knows as the Sequential Probability Ratio Test (SPRT). In this framework, the likelihood function is incrementally updated after every new sample arrival [5], given* $x^n = \{X_1, \cdots, X_n\}$*:*

$$\mathbf{L}(x^n) = \log \frac{f_1(x^n)}{f_0(x^n)} = \mathbf{L}(x^{n-1}) + \log \frac{f_1(X_n)}{f_0(X_n)} \tag{4.9}$$

where reject H_0 *if* $\mathbf{L}(x^n) > \alpha$ *and reject* H_1 *if* $\mathbf{L}(x^n) < \beta$*, where* α *and* β *are two constants. If* $\alpha \geq \mathbf{L}(x^n) \geq \beta$*, we continue to accept new samples till a decision can be made.*

SPRT simulates the way human makes decisions. One makes decision if one has enough confidence and will continue to receive information if not. In SPRT, we do not need to pre-determine the size of the test. Instead, the size is adaptively determined based on the observations. Even better is that SPRT requires fewer samples than standard non-sequential N-P test given the same FP Rate constraint. The expected number of samples for certain FP rate α is given as [60]

$$\begin{cases} \mathbf{E}(N|H_0, H_1) & \approx \frac{\log(\alpha)}{D(f_0|f_1)} \quad \text{for Sequential} \\ \mathbf{E}(N|H_0, H_1) & \approx \frac{\log(\alpha)}{C(f_0|f_1)} \quad \text{for Non-sequential} \end{cases} \tag{4.10}$$

in which $D(f_0|f_1)$ is the Kullback-Leibler (K-L) distance and $C(f_0|f_1)$ the Chernoff distance. For Gaussian variable, the K-L distance is usually greater than Chernoff distance. Therefore, SPRT needs fewer samples to reach a decision.

The optimality of the sequential test motivates us to formulate the power disaggregation problem based on it.

Now we move on to multiple-hypothesis test. If we have one null hypothesis and k alternative hypotheses, from [7], we should compare one hypothesis with all the other choices. Suppose that the j^{th} hypothesis has a prior π_j, we can write the posterior probability of the j^{th} hypothesis as:

$$p_n^j = \frac{\pi_j \prod_{i=1}^n f_j(X_i)}{\sum_{j=0}^k \pi_{j'} \prod_{i=1}^n f_{j'}(X_i)} \tag{4.11}$$

For computation purpose, we use its inverse as the test statistic. Decision is made towards the j^{th} hypothesis if the threshold corresponding to the j^{th} hypothesis, which is noted as χ_j, is exceeded. Otherwise, more data are sampled:

$$F_1^n(j) = \frac{1}{p_n^j} < \chi_j \tag{4.12}$$

The algorithm works as in Figure 4.7(a), in which $F_1^n(1)$ exceeds the threshold, whereas $F_1^n(2)$ goes to the opposite direction. The threshold for the j^{th} hypothesis is calculated as $\chi_j = \alpha \left(\delta_j \sum_{j'} \frac{\pi_{j'}}{\delta_{j'}} \right)^{-1}$, in which $\delta_j = \min_{j' \neq j} D(f_j | f_{j'})$ [2]. The number of samples we need to reach a decision is:

$$n = \inf \{ n \geq 1, F_1^n(j) < \chi_j, \forall j \} \tag{4.13}$$

The second issue is to locate the edge efficiently. Usually, exact edge location is not known *a priori*. If we assume the edge is at time τ. Then, the accumulation of the probability ratio functions in Equ. (12) will start from τ, and the number of sample n will be τ dependent:

$$n(\tau) = \inf \left\{ t \geq 1, S_\tau^t(j) < A_j, \forall j \right\} \tag{4.14}$$

As we have discussed before, the functions $F_\tau^t(j)$ will only move toward threshold when its hypothesis is the truth. Thus, if a guess is ahead of the true location, the function will move away from threshold for a while; whereas if the guess is behind the true location, the function will have a late hit to the threshold, as shown in Figure 4.7(b). Therefore, the exact location will be determined by the function that *firstly* hit the threshold, as:

$$n = \inf_\tau n(\tau)$$

For Gaussian distribution, the density decays very quickly for outliers. This is not preferable from a numerical standpoint. The log-likelihood function is more promising. Thus, the original formulation is modified as follows:

$$\begin{aligned} N(\tau) &= \inf_{t \geq 1} \left\{ F_\tau^t(k) < \chi_k, \forall k \right\} \\ &\approx \inf_{t \geq 1} \left\{ \max_{j \neq k} \sum_{i=t-\tau}^{t} \log \frac{f_j(X_i)}{f_k(X_i)} < \log \frac{\chi_k}{k}, \forall k \right\} \\ &\approx \inf_{t \geq 1} \left\{ \sum_{i=t-\tau}^{t} \max_{j \neq k} \log \frac{f_j(X_i)}{f_k(X_i)} < \log \frac{\chi_k}{k}, \forall k \right\} \end{aligned} \tag{4.15}$$

The first approximation is to relax the left side of the inequality and transform it into log-likelihood ratio, while the second puts the maximum inside the sum and takes the maximum at each step, hence will make the test robust to noisy data (i.e. "spikes" that frequently appear in power stream data).

The MSPRT originates from the Edge-based model. However, by sequentially considering the density function, MSPRT borrows ideas from the probabilistic HMM and it appears that it combines some of their advantages.

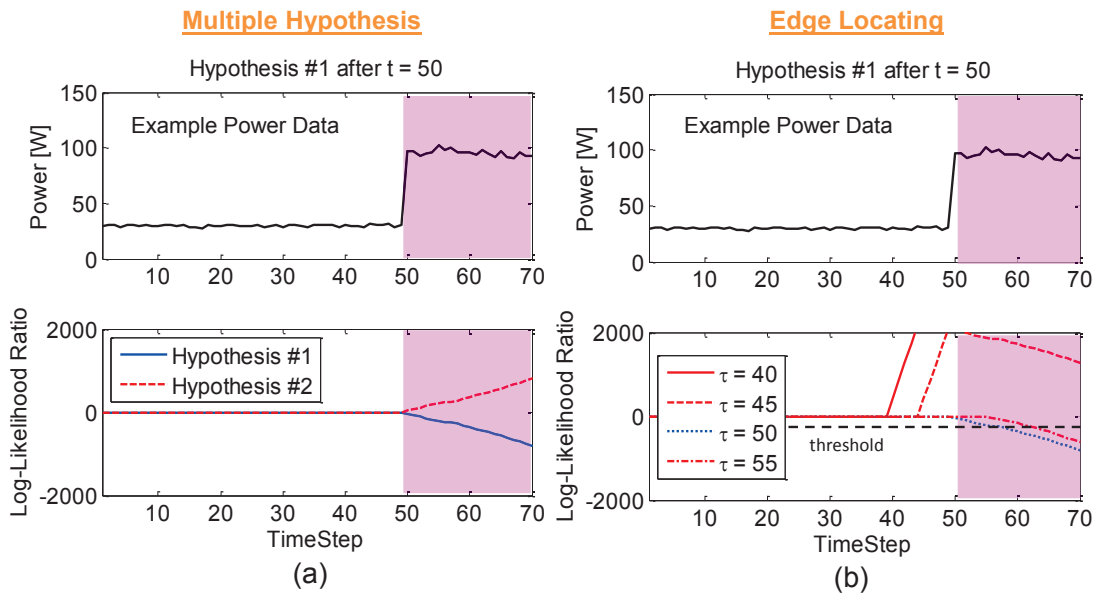

Figure 4.7: Demonstration of MSPRT: (a) Log-likelihood function evolution; (b) Edge positioning

Sequential Test Based Power Disaggregation: Results

The k-hypotheses in MSPRT can be used to test the status of k different appliances. By sequentially applying MSPRT to the power stream, we can find the right hypothesis, hence the right switching appliance. Thus, MSPRT can be used in power disaggregation applications. We will discuss this more in this section and compare MSPRT with HMM and the Edge-Based Model.

It is also worth noting that to use MSPRT in the power disaggregation application; we need to know in advance the appliance profiles that connect to the sensor node. This is usually done by learning from a period of ground-truth data. Apart from that, MSPTR does not ask for extra parameters compared to that of HMM or Edge-based Model. For the situation in which some appliances can have multiple states, these states can be transformed into virtual appliances, which presents a similar problem as before.

Pseudo-realistic power stream is used in this study's analysis. Firstly, a set of real data was collected by measurement. Several meters have been deployed in 550 Cory Hall at UC Berkeley collecting power streams of plug-in loads. Each appliance has its characteristics profile, and some appliances, such as a laptop computer, have

a non-stationary pattern, as illustrated in Figure 4.8.

The data collected by measurement has limited stochasticity, so white Gaussian noise and/or impulse noise was added to introduce randomness. By tuning the noise parameters, the potential performance limit of different methods can be benchmarked. Thirty Monte Carlo simulations were performed at each setting of parameters.

To evaluate the model performance, one criteria we used is the Detection Error Rate (DER), which is the gap between the detected and the true number of edges, i.e.:

$$\text{DER} = \frac{n_{detect} - n_{true}}{n_{true}} \tag{4.16}$$

Another one we used is the LDA score, or F-score [41]. LDA score integrates the Precision and Recall scores. Precision is given by

$$\text{Prec} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

in which TP is True Positive rate, FP is False Positive rate. Recall is given by:

$$\text{Rec} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

in which FN is False Negative rate. The LDA is eventually given by:

$$\text{LDA} = \frac{2\text{Prec} \times \text{Rec}}{\text{Prec} + \text{Rec}} \tag{4.17}$$

Therefore, the efficacy of the various methods will be judged in terms of achieving low DER and high LDA values.

In this study's simulation, one desktop computer, one computer monitor, and one laptop computer were included, as these are the most common appliances in a typical office building. We also included a water heater with a pump for water filtering. The patterns for the five appliances are shown in Figure 4.9. Note that non-stationary time series is also considered here (e.g., in the left figure). Non-stationarity definitely bring about extra challenge, and in this work, it was handled by considering the dynamic time series model.

There are two groups of study in this section. In the first group we only consider Gaussian random noise, and the data is modeled as $p_t = h(\mathbf{s}_t) + z_t$ with $h(\mathbf{s}_t)$ being the state-dependent clean signal, and z_t being the Gaussian noise with variance σ_z^2. The impact of noise is investigated by tuning σ_z^2 from 1 to 256, based on the measurements. The state duration is modeled as Gamma distributed [41], and it was assumed that at each step, one appliance switches at most.

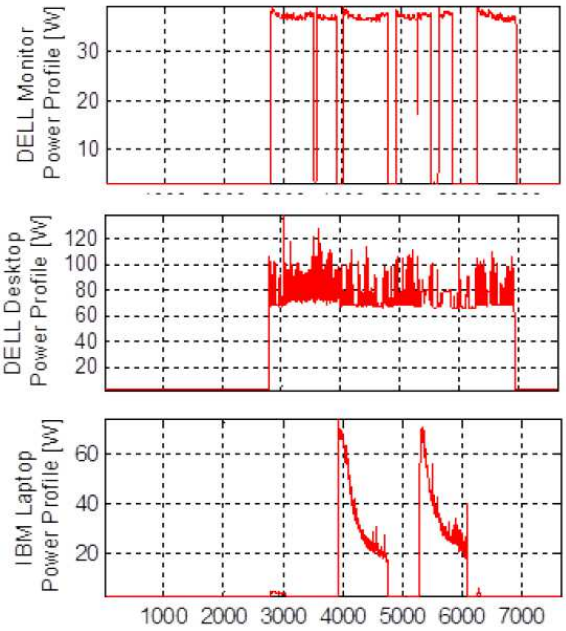

Figure 4.8: Measured power profile of desktop, monitor and laptop

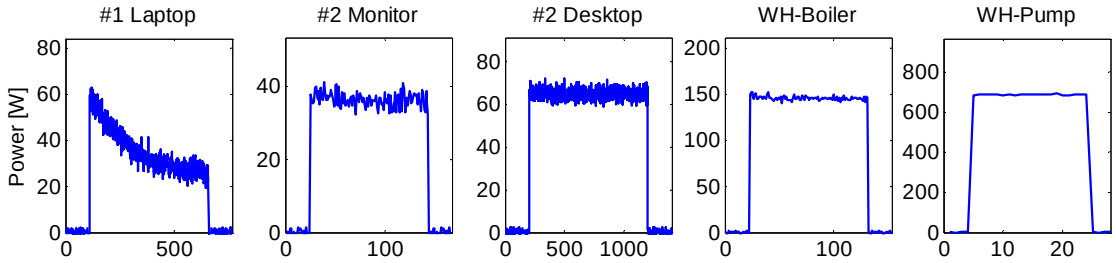

Figure 4.9: Simulated power pattern for five devices

The three methods under study in this section are as follows: the MSPRT, the HMM, and the Edge-Based Model. The simulation results for these three methods are summarized by showing the LDA in Figure 4.10, and the DER in Figure 4.11.

In terms of LDA for the laptop and monitor, there is a drop in LDA above a certain noise level for the Edge-Based Model. For fixed sample detection, the expected number of samples needed is following equation (4.10). If this number is over the test sample size (which increases as the noise level increases), then the changes could be missed. MSPRT adaptively learns the test samples size, and HMM

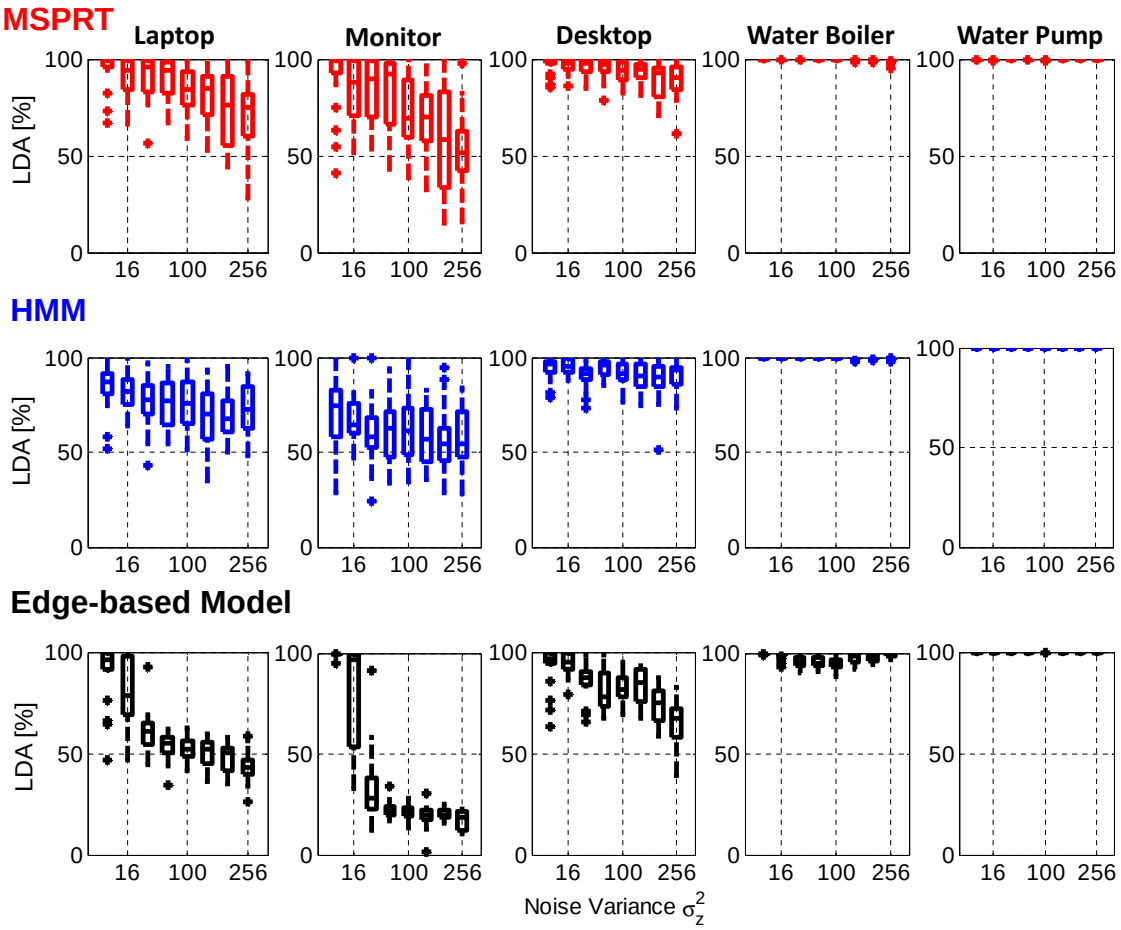

Figure 4.10: Monte Carlo Simulated LDA results for the five appliances as a function of Gaussian noise amplitude, under the three models

tunes itself by introducing state transitions. Thus, they do not have the abrupt drop in LDA, as shown in Figure 4.10, though MSPRT is slightly better.

In terms of DER, MSPRT is the most accurate method, since the state changes only after the edge is detected and the sample size can be self-tuned. The edge-based model suffers a sudden increase of DER at high noise levels because it is non-sequential, whereas HMM is worse in DER compared to MSPRT, since the state stickiness is not well modeled in HMM.

The impact of impulse noise was studied in the second group. Here, we model the data as $p_t = h(\mathbf{s}_t) + z_t + \lambda w_t$, where w_t is the impulse noise term with variance $\sigma_w^2 \gg$

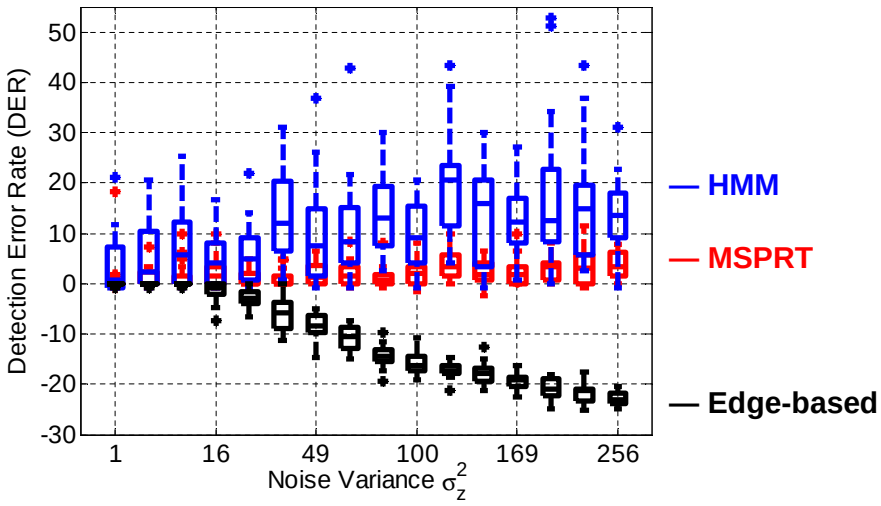

Figure 4.11: Monte Carlo simulated DER as a function of Gausian noise amplitude for the three methods under study

σ_z^2, and $\lambda \in 0, 1$ is a Bernoulli process that models the impulse noise probability. The impact of impulse noise was investigated by varying noise variance σ_w^2 as well as the Bernoulli process probability $\Pr(\lambda = 1)$. Based on measured data, the range of σ_w^2 was set from 50^2 to 150^2, and $\Pr(\lambda = 1)$ was set to be from 0.02 to 0.5. The only focus here is on MSPRT and HMM, since these methods give better average performance.

The LDA and DER of MSPRT and HMM are shown in Figure 4.13, Figure 4.14 and Figure 4.15. They have similar performance in terms of LDA, and MSPRT, not surprisingly, has better DER than HMM. However, even for MSPRT, the DER goes beyond 100% as noise-level increases.

It is well known that tests assuming a Gaussian distribution are sensitive to outliers or impulses [7]. In the presence of impulse noise, both MSPRT and HMM suffer from degradation caused by the outliers. Therefore, it is necessary to introduce a robust model. This is found to be most efficient for MSPRT.

Several distributions can model data sets that either have longer-than-Gaussian tails, or are skewed. Examples include the student t-distribution or the Gamma distribution. In this work, inspired by the Huber Robust Loss Function [32], a robust distribution that has quadratic decay in its main body and linear decay towards its tails is introduced. Assuming, without loss of generality, that the data is zero-

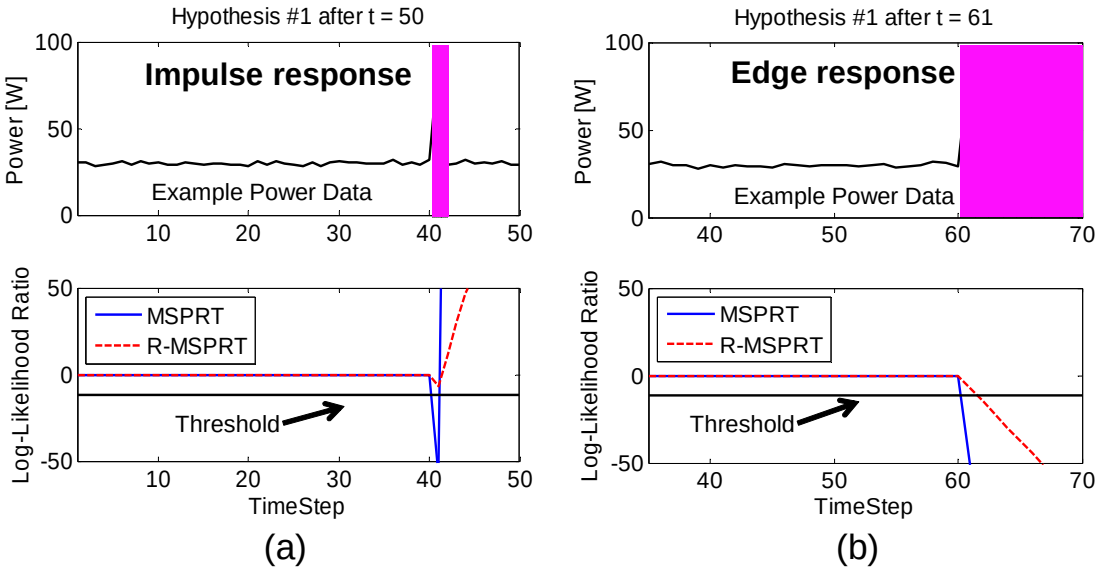

Figure 4.12: Demonstration of RMSPRT: (a) Impulse noise response and (b) True edge response

centered and standardized ($y = \frac{x}{\sigma_k}$):

$$\log f_k \cong -\frac{y^2}{2}\mathbb{1}\left\{|y| \leq \xi\right\} - \frac{|y| + \xi^2 - \xi}{2}\mathbb{1}\left\{|y| > \xi\right\}$$

The normalization coefficient of f_k can be obtained as:

$$C = 2\sigma_k \left\{\sqrt{2\pi}\left(\Phi(\xi) - 0.5\right) + 2e^{-\frac{1}{2}\xi^2}\right\} \propto \sigma_k$$

in which $\Phi(\xi)$ is the cumulative density function (CDF) of the standard Normal Distribution. Thus, the log-likelihood function can be written similar to the Gaussian case ($y_{k(j)} = \frac{x}{\sigma_{k(j)}}$) as $\log \frac{f_j}{f_k}$.

A demonstration of the Robust MSPRT (R-MSPRT) is shown in Figure 4.12. From Figure 4.12(a), R-MSPRT is less sensitive to impulse noise. However, as seen in Figure 4.12(b), R-MSPRT is, at the same time, less likely to detect *true* changes, even in a normal setting. We need to pay attention to this tradeoff as we choose the parameters.

We compare the performance of this R-MSPRT with the MSPRT and the HMM in Figures 4.13 to Figure 4.15, and we only focus on the first three appliances in

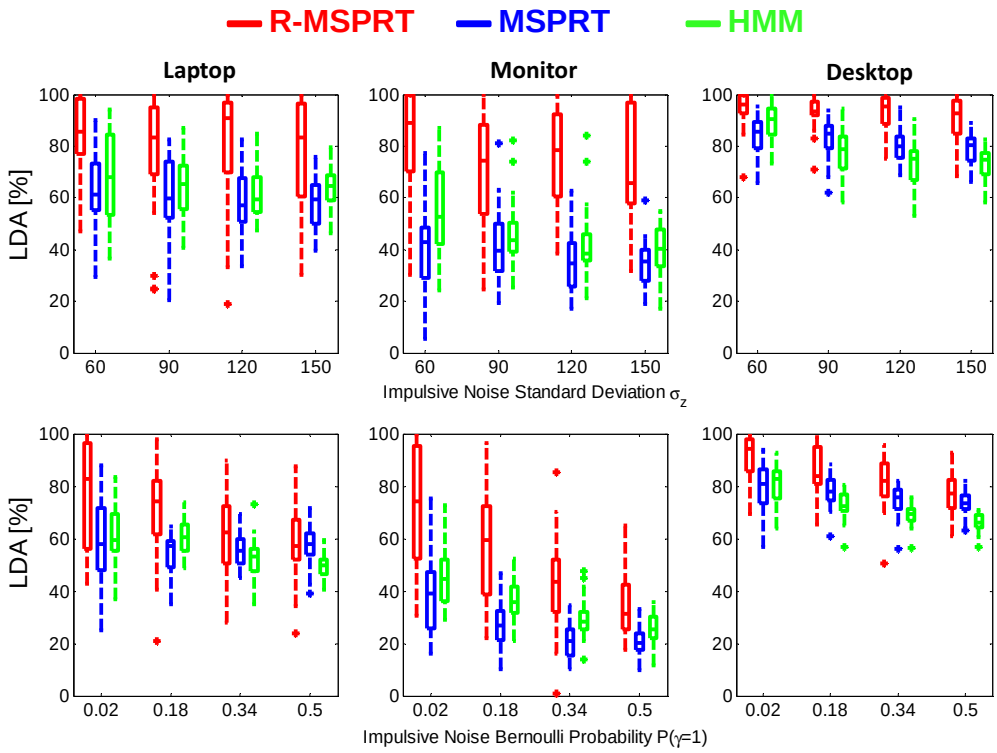

Figure 4.13: LDA as a function of impulse noise amplitude and impulse Bernoulli probability for the first three appliances under study, using Monte Carlo simulated data

Figure 4.9. The R-MSPRT gives better LDA compared with the other two methods, and it shows much better DER as well. Actually, R-MSPRT has DER consistently below 5% and does not suffer from much degradation as noise variance increases. This is due to the introduction of a noise that is robust to large deviation. It should be noted that R-MSPRT has similar computational complexity to the ordinary MSPRT.

A problem of R-MSPRT is that when the observed data is ambiguous, many samples may need to be processed in order to satisfy the confidence requirement. A decision can be made before a certain number of samples are reached by a truncated SPRT [60], which could be a subject for future study.

With power disaggregation techniques, the ON/OFF states of individual appliances can be obtained, and then ON/OFF probabilities used in simulation may be calculated. In the next section, we will test the performance of our model in Chapter 3 in real buildings.

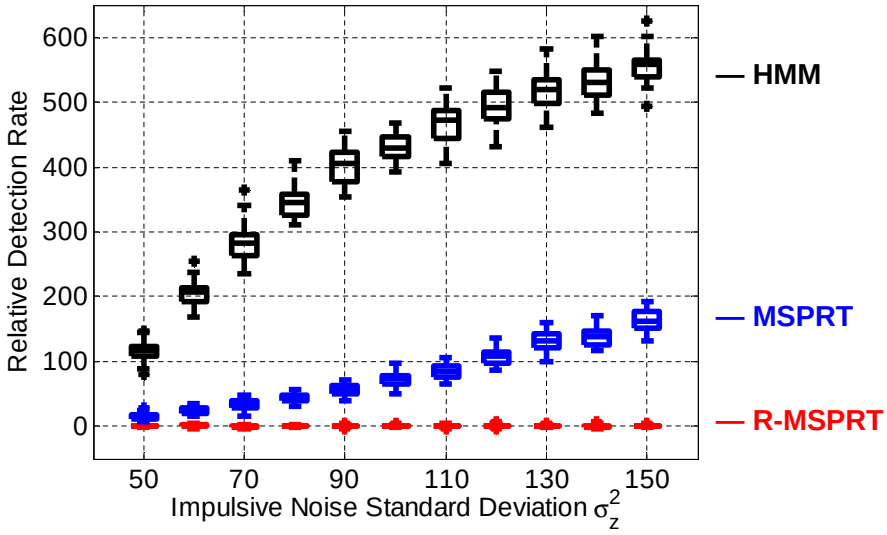

Figure 4.14: Monte Carlo simulated DER as a function of Bernouli noise probability showing the efficacy of the Robust noise model

4.3 Experiments and Results

Model Setting

To make it clear, the *Bottom-up* approach is built based on the following steps.

- Firstly, the ON/OFF states of appliances are extracted using the power disaggregation algorithm discussed in Section 4.2.
- Secondly, the *Statistical Parameters* are extracted as illustrated in Section 3. The shared appliances are modeled as Non-Homogeneous Poisson Process (NHPP).
- Thirdly, the *Field Parameters* are also extracted either in a *Level-III* or in a *Level-II* model. For accuracy reasons, the *Level-I* model is not considered in this work.

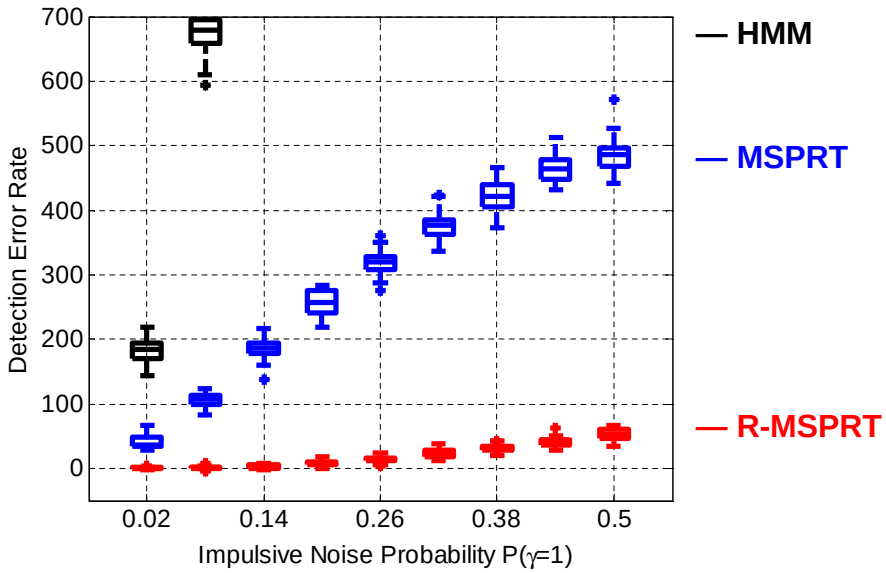

Figure 4.15: Monte Carlo simulatede DER as a function of Bernouli noise amplitude showing the efficacy of the Robust noise model.

Model Training

The model was trained based on ACme power-meter readings during the 2014 Fall Semester (i.e., 09/01/2014 to 12/01/2014) at UC Berkeley's CREST Center. The ACme meters are implemented at an individual level, and power disaggregation technique was used to decompose the aggregated cubicle level to an appliance level, as discussed in Section 4. Then, following the rules in Section 3, a *Bottom-Up* model based on the stochastic ON/OFF probability was built. As background information, there are 18 occupants in the CREST space with 3 desktops, 10 monitors, and 11 laptops.

Cory Hall 406 Winter Semester

In the 2014âĂŞ15 winter semester (12/28/2014 to 01/16/2015), the CREST space is much less occupied, with only 3 monitors, 5 laptops, and 2 desktops actively running. The simulated and measured mean and standard deviation of the power consumption of the *Level-III* model are shown in Figure 4.16. Most of the levels are captured, and the error in standard deviation comes from the limited data in our study, especially

for *desktop*. For the *Level-III* model, much field information is needed, which in most cases is unreliable. For a new test space, the *Level-II* model is preferred.

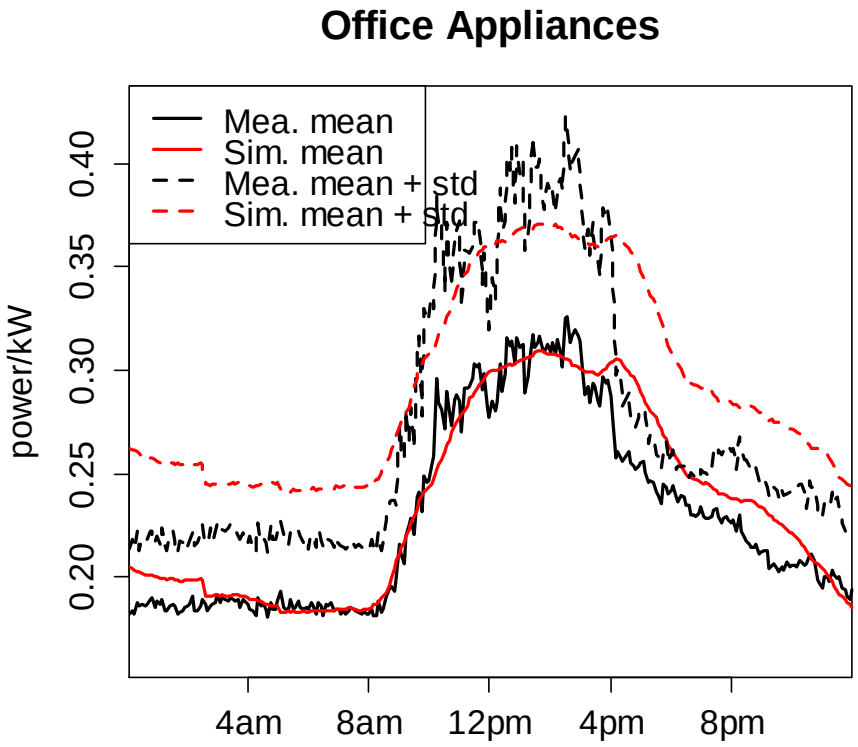

Figure 4.16: The simulated (Sim.) and measured (Mea.) mean and standard deviation (std.) of the power consumption (in kW).

Sutardja-Dai Hall 4th Floor Fall Semester

The second test was carried out on the fourth floor of UC Berkeley's Sutardja-Dai Hall. The schematic of the floor space is illustrated in Figure 4.17. There were 62 occupants on fourth floor, with 46 of them in cubicles and 16 of them in offices. There are also three printers and one kitchen.

The second test is for *Level-II* model, and the difference of *Level-II* model from *Level-III* model is that it only takes the number of occupants as input and infer

the number of appliances based on the *possession* probability. This probability is estimated from the fourth floor of Sutardja-Dai Hall, and CREST center space, as well as SWARM lab in Cory Hall, all at UC Berkeley. There are more than 110 users included. For each individual, the probability of having an laptop is 0.4545, monitor is 0.6545, and the desktop is 0.2455. Printers are special, for office users, each occupant has a printer; for cubicle users, each common space has roughly one printer. Definitely, it should be noted that the computer appliances are still the major power consumption.

Figure 4.17: Schematic of CITRIS fourth floor.

Whole-building plug-in measurements were collected during 09/01/2014 to 12/01/2014, and a *Level-II* model simulation was completed. The result is shown in Figure 4.18. Most of the daytime variation is captured but with an unidentified baseline missing. This baseline is almost constant and is believed to correspond to the constant server or processor operation on this floor. Thus, for the *Level-II* model, adding the number of processors into the model, apart from the number of occupants, will probably yield results that are more accurate. However, it should also be noted that such processors could be task-specific.

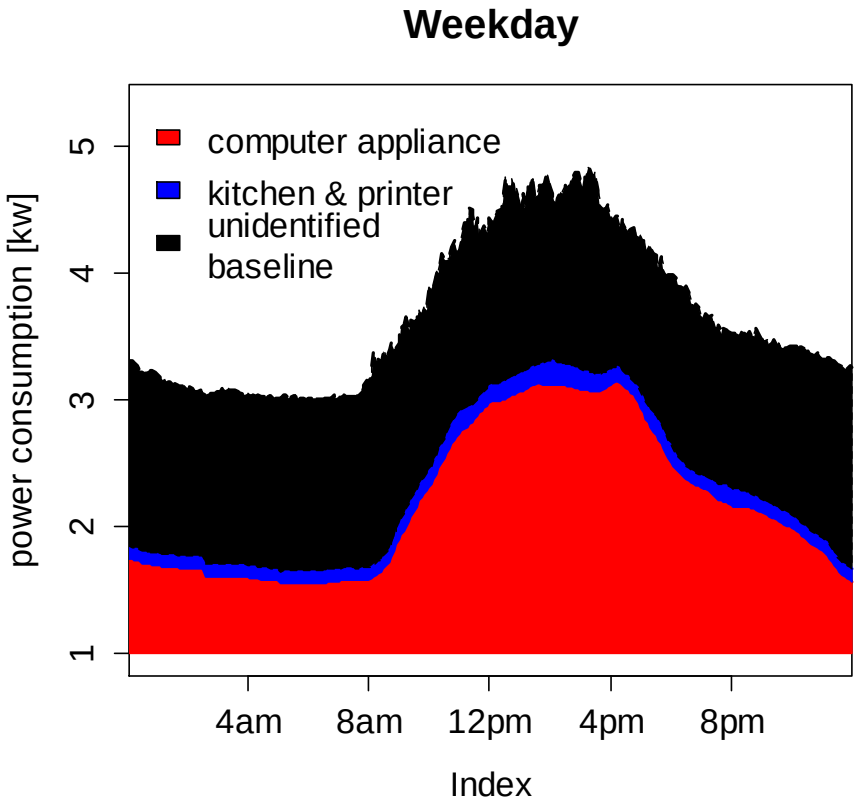

Figure 4.18: Simulated and measured data from CITRIS fourth floor. The unidentified baseline is the measurement minus the simulation.

4.4 Conclusions and Future Tasks

In this chapter, based on the study from Chapter 3, we make use of the *Bottom-up* approach to model the building plug-in loads power consumption under different scenarios.

Power disaggregation as an important technique to filter appliance ON/OFF state from aggregated raw power sequences is discussed. A new disaggregation technique based on multiple-hypothesis sequential testing and robust statistics is introduced, showing stable performance under impulsive power sequences.

The experiment was then conducted in Cory Hall and Sutardja-Dai Hall at UC Berkeley. The model demonstrates a strong capability to simulate seasonal and

daytime variation of power consumption in commercial buildings.

For the next step, attention could be given to the modeling and feasibility analysis of a *Level-I* model with simplicity and commercial potential.

Chapter 5

Bottom-Up End-Use Monitoring: A Dimensionality Reduction Approach

5.1 Introduction

In Chapter 3 and Chapter 4, the *modeling* issue in *Bottom-up* approach was discussed. In this chapter, we will move on to study the *monitoring* issue. As discussed before, the advantage of *Bottom-up* approaches is its coverage of fine-grained individual power consumption. However, as in other multivariate systems, when the amount of data scales up, several challenges arise in the efficiency of monitoring, storage, and the performance of statistical learning algorithms [31]. By providing a more efficient lower-dimensional reconstruction of the original system, dimensionality reduction[1] is one of the techniques that can help to overcome these issues [58].

Among the dimensionality reduction techniques, Principal Component Analysis (PCA) is most widely known. PCA finds the linear projection of the original data matrix that explains the largest portion of the variance, known as the Principal Component (PC). However, when data are not consistently Gaussian distributed[2], the linear projected Principal Component is usually not interpretable. For example, when data are binary, which happens a lot in behavioral science, the linear projection is usually not binary anymore.

Recently, a generalized PCA framework for exponential-family distributed data is developed (also known as the *e*PCA) [12]by formalizing PCA into a generalized low-rank approximation framework. In the case of Bernoulli random variables, the

[1]Dimesionality reduction, dimension reduction, dimensional reduction refer to the same thing, in this work.

[2]By consistency the streaming data are following same distribution.

generalized PCA is called Logistic PCA (LPCA).

Moreover, with the explosion of streaming data nowadays, it is also important to have the algorithm applicable in real-time setting. Running batch mode LPCA every time when new data point comes in is definitely too costly, and a sequential version of LPCA would be highly preferred.

In this chapter, we will study the LPCA mentioned before on multivariate binary data and extend it to a sequential version called SLPCA, based on the sequential convex optimization theory [80] [64]. The convergence property of this algorithm is discussed. An application in building energy end-use profile modeling is investigated based on this method.

This chapter is organized as follows: In Section 5.2, the background and the detail of the algorithm is given, including PCA, exponential family, and eventually the sequential LPCA (i.e. SLPCA) which we propose. In Section 5.3, the convergence property of the algorithm is discussed, followed by the simulation results as well as the application in energy end-use modeling in Section 5.4. In Section 5.5, conclusion is drawn.

5.2 Algorithm Framework

PCA as a dimensional reduction technique has been well studied, and our Sequential LPCA is essentially a generalized incremental version of the classical model.

Principal Component Analysis

PCA is a well-known technique for dimensional reduction for high dimension data. It is of special importance in high dimensional regression model, and in a variety of applications, ranging from face recognition to generalized machine learning [70] [31].

Apart from the maximum variance projection perspective mentioned before, there is another perspective of PCA called the low-rank factorization perspective [68]. Let $\mathbf{X} \in \mathbb{R}^{n \times p}$ be p-dimensional data with length n. PCA finds a lower rank matrix $\mathbf{\Theta}$ to minimize certain loss function. In conventional PCA, the loss function is in Frobenious norm (or squared-error) shape:

$$\min_{\mathbf{\Theta}} \|\mathbf{X} - \mathbf{\Theta}\|_F^2 \tag{5.1}$$

in which $\|\cdot\|_F$ is the Frobenious norm. The lower rank matrix Θ will contain the principal components (PCs).

When $\mathbf{X} \in \mathbb{R}^{n \times p}$ follows Gaussian distribution, this minimization problem is essentially maximum likelihood low rank reconstruction [68]. From this standpoint,

if X follows other distribution, we can also extract the PCs by designing the loss function as negative likelihood function $L(\mathbf{X}\|\mathbf{\Theta}) = -\log \Pr(\mathbf{X}\|\mathbf{\Theta})$. However, there are two issues needs to be address.

- The maximum likelihood low rank reconstruction problem is not always as straightforward to solve as in equation (5.1). For non-convex loss function, global optimal solution is not guaranteed.
- As illustrated before, the low rank reconstructed matrix $\mathbf{\Theta}$ need to be consistent with the original distribution.

Fortunately, when original data $\mathbf{X} \in \mathbb{R}^{n\times p}$ follows Exponential Family distribution, the two issues above can be tackled.

Exponential Family

Definition 5.2.1 (Exponential Family)**.** *In the exponential family of distributions the conditional probability of a value X given parameter value Θ takes the following form:*

$$\log P(X|\Theta) = \log P_0(X) + X\Theta - G(\Theta) \tag{5.2}$$

in which, Θ is called the natural parameter of the distribution. Then we have $\mathrm{E}[X] = \nabla G(\Theta) = g(\Theta)$ is the inverse canonical link function, and $\mathrm{Var}[X] = \nabla\nabla^T G(\Theta)$.

- Log-likelihood function of exponential family distribution is concave with respect to the natural parameter Θ, hence the negative likelihood minimization is efficient.
- Since $E[X] = g(\Theta)$, we can interpret the Principal Components as $g(\Theta)$.

Example 5.2.1. *In the case of Gaussian distribution, the negative log-likelihood follows*

$$L(x\|g(\theta)) = \frac{1}{2}(x-\theta)^2$$

It coincides with the Frobenious norm function in equation (5.1)*.*

Example 5.2.2. *In the case of Bernoulli distribution, the negative log-likelihood is the logit function*

$$L(x\|g(\theta)) = \log(1+\exp(-x^*\theta)) \tag{5.3}$$

where $x^* = 2x-1 \in \{-1,1\}$. In this case, the loss function is a convex function of θ. However, the minimum could be at infinity. Hence, usually we put a regularization

term $\gamma\frac{\theta^2}{2}$ *there, with the full loss function being* $L(x\|g(\theta)) = \log(1+\exp(-x^*\theta))+\gamma\frac{\theta^2}{2}$. *Note that for Bernoulli distribution, the inverse canonical link function is* $g(\theta) = \frac{1}{1+\exp(-\theta)}$ *with* 0 *and* 1. *Thus, the Principal Components can be interpretable.*

Exponential Family PCA

In this work, we will only work on Bernoulli variable, as in the second example, we replace Frobenious loss in (5.1) by the logit function. For multivariate binary date matrix $\mathbf{X} \in \mathbb{R}^{n\times p}$, we have:

$$L(\mathbf{X}\|\mathbf{\Theta}) = \sum_{i,j} \log(1 + e^{-x^*_{ij}\theta_{ij}}) \tag{5.4}$$

For a rank-r matrix $\mathbf{\Theta}$, we can always write it as a product of two matrices $\mathbf{\Theta} = \mathbf{A}\mathbf{V}^T$ where $\mathbf{A} \in \mathbb{R}^{N\times r}$ and $\mathbf{V} \in \mathbb{R}^{P\times r}$, both rank-$r$. Thus equation (5.4) becomes:

$$L(\mathbf{X}\|g(\mathbf{A}\mathbf{V}^T)) = \sum_{i,j} \log(1 + e^{-x^*_{ij}(\mathbf{A}\mathbf{V}^T)_{ij}}) \tag{5.5}$$

The optimization problem in (5.5) is not jointly convex because of the $\mathbf{A}\mathbf{V}^T$ term. However, interestingly, from some mathematical discussions [53][1][26], every local minimum is a global minimum, which is partially because of the interchangeability between $\mathbf{A}$ and $\mathbf{V}$. Local minimum can be obtained from alternating project method, which means that we solve $\mathbf{A}$ with $\mathbf{V}$ fixed, and then solve $\mathbf{V}$ with $\mathbf{A}$ fixed, and iterate this process:

$$\begin{cases} \mathbf{A}^t &= arg\min\limits_{\mathbf{A}\in\mathbb{R}^{n\times r}} L(\mathbf{X}\|g(\mathbf{A}(\mathbf{V}^{t-1})^T)) + \frac{\gamma}{2}\|\mathbf{A}\|_F^2 \\ \mathbf{V}^t &= arg\min\limits_{\mathbf{V}\in\mathbb{R}^{p\times r}} L(\mathbf{X}\|g(\mathbf{A}^t\mathbf{V}^T)) + \frac{\lambda}{2}\|\mathbf{V}\|_F^2 \end{cases} \tag{5.6}$$

in which $\frac{\gamma}{2}\|\mathbf{A}\|_F^2$ and $\frac{\lambda}{2}\|\mathbf{V}\|_F^2$ are regularization terms.

Equation (5.5) is marginally convex for both $\mathbf{A}$ and $\mathbf{V}$, hence each equation in (5.6) is convex and can be solved efficiently by Newton's method. Without loss of generality, we mark the local minimum obtained from (5.6) as $\mathbf{A}^*$ and $\mathbf{V}^*$, and this solution is called Batch Logistic PCA (BLPCA) solution.

Sequential Logistic PCA (SLPCA)

As we work with streaming data (n is not fixed), $\mathbf{A} \in \mathbb{R}^{N\times r}$ changes in size as n increases, though the dimension of $\mathbf{V}$ is still fixed. It would be too costly to update the whole $\mathbf{A}$ matrix each time when we have a new data point.

Conventionally as we have accumulated loss functions $L(w) = \sum_t L_t(w)$ and w fixed in size, we can make use of gradient descent to update w sequentially:

$$w^t = w^{t-1} - \eta \Delta L_t(w^{t-1}) \tag{5.7}$$

However, $\mathbf{A}$ matrix in our case is not fixed in size. Hence, we choose to do a further approximation. At each step t when a new data comes in, we only look at the t-th row of $\mathbf{A}$, which we note by $\mathbf{a}_t$, $t = 1, \cdots, n$. Since the loss function in equation (5.5) can be decomposed by the summation of a loss function of each row of $\mathbf{A}$, we only optimize over the loss functions relevant to that row ($\mathbf{a}_t$), called $L_t(\mathbf{a}_t, \mathbf{V})$:

$$L_t(\mathbf{a}_t, \mathbf{V}) = L(\mathbf{x}_t \| g(\mathbf{a}_t \mathbf{V}^T)) = \sum_j \log(1 + e^{-x^*_{tj}(\mathbf{A}\mathbf{V}^T)_{tj}}) \tag{5.8}$$

and note that the total loss function is the aggregation of (5.8):

$$L(\mathbf{x}_t \| g(\mathbf{A}\mathbf{V}^T)) = \sum_t L_t(\mathbf{a}_t, \mathbf{V})$$

This method is similar to [54]. At each time t, instead of working on the full $\mathbf{A}$ up to step t, we only solve for the current element $\mathbf{a}_t$. We mark the solution as $\widetilde{\mathbf{a}}_t$. As for $\mathbf{V}$, at each step we optimize it over all the row-level loss functions up to t, and we mark the solution as $\widetilde{\mathbf{V}}^t$).

In this algorithm, for $t = 1, \cdots, n$:

$$\begin{cases} \widetilde{\mathbf{a}}_t & = arg \min_{\mathbf{a} \in \mathbb{R}} L_t(\mathbf{a}, \widetilde{\mathbf{V}}^{t-1}) + \frac{\gamma}{2} \|\mathbf{a}\|_F^2 \\ \widetilde{\mathbf{V}}^t & = arg \min_{\mathbf{V} \in \mathbb{R}^p} \sum_{s=1}^t L_s(\widetilde{\mathbf{a}}_s, \mathbf{V}) + \frac{\lambda}{2} \|\mathbf{V}\|_F^2 \end{cases} \tag{5.9}$$

The one for $\widetilde{\mathbf{a}}_t$ in (5.9) is easy to solve with a Newton's method. The one for $\widetilde{\mathbf{V}}^t$ in (5.9) deal with a target function increasing in size. However, we can still make use of the stochastic gradient descent method as in equation (5.7).

$$\widetilde{\mathbf{V}}^t = \widetilde{\mathbf{V}}^{t-1} - \eta_t \nabla_{\mathbf{V}} L_t(\widetilde{\mathbf{a}}_t, \widetilde{\mathbf{V}}^{t-1}) \tag{5.10}$$

where η_t is the step size. The choice of step size η_t deserves some discussions.

This method is called Sequential LPCA (SLPCA), and we will investigate the convergence property of this algorithm in the next section. The full SLPCA algorithm is shown below in Algorithm 1.

begin
  Input: data $\mathbf{X} \in \mathbb{R}^{n \times p}$, $\mathbf{X}^* = 2\mathbf{X} - 1 \in \{-1, 1\}$;
  Initialize: $\widetilde{\mathbf{V}}^t \approx 0, C, \gamma, \epsilon, \beta \in (0,1), \alpha$;
  for $t = 1, \ldots, n$, $l_t(\widetilde{\mathbf{a}}_t) \doteq L_t(\widetilde{\mathbf{a}}_t, \widetilde{\mathbf{V}}^{t-1}) + \lambda \frac{\|\widetilde{\mathbf{a}}_t\|_F^2}{2}$ **do**
    Set $\widetilde{\mathbf{a}}_t = 0$, $\Delta = \nabla l_t(\widetilde{\mathbf{a}}_t) \left(\nabla^2 l_t(\widetilde{\mathbf{a}}_t)\right)^{-1} \nabla l_t(\widetilde{\mathbf{a}}_t)$;
    while $\lambda > \epsilon$ **do**
      Let $\Delta = -\left(\nabla^2 l_t(\widetilde{\mathbf{a}}_t)\right)^{-1} \nabla l_t(\widetilde{\mathbf{a}}_t)$, $d = d_0$;
      while $l_t(\widetilde{\mathbf{a}}_t + d\Delta) > l_t(\widetilde{\mathbf{a}}_t) + \alpha d \nabla l_t^T \Delta$ **do**
        Update $d = \beta d$;
      end
      Update $\widetilde{\mathbf{a}}_t = \widetilde{\mathbf{a}}_t + d\Delta$;
      Update $\Delta = \nabla l_t(\widetilde{\mathbf{a}}_t) \left(\nabla^2 l_t(\widetilde{\mathbf{a}}_t)\right)^{-1} \nabla l_t(\widetilde{\mathbf{a}}_t)$;
    end
    Set η_t;
    Update $\widetilde{\mathbf{V}}^t = \widetilde{\mathbf{V}}^{t-1} - \eta_t \nabla_{\mathbf{V}} L_t(\widetilde{\mathbf{a}}_t, \widetilde{\mathbf{V}}^{t-1})$
  end
end

Algorithm 1: Sequential LPCA (SLPCA) Pseudo-Code

5.3 Convergence Analysis

In this section, we will study the convergence of SLPCA with respect to BLPCA algorithm in terms of some widely-used settings from online statistical learning society.

Evaluation Settings

- *Batch Loss Function* (BLF), use $\{\mathbf{A}^*\}$ $\{\mathbf{V}^*\}$:

$$\text{BLF} = \frac{1}{n} \sum_{t=1}^{n} L_t(\mathbf{a}_t^*, \mathbf{V}^*) \tag{5.11}$$

- *Sequential Loss Function* (SLF), use $\{\widetilde{\mathbf{a}}_t\}$ $\{\widetilde{\mathbf{V}}^n\}$:

$$\text{SLF} = \frac{1}{n} \sum_{t=1}^{n} L_t(\widetilde{\mathbf{a}}_t, \widetilde{\mathbf{V}}^n) \tag{5.12}$$

- *Regret Loss Function* (RLF), use $\{\widetilde{\mathbf{a}}_t\}$ $\{\widetilde{\mathbf{V}}^t\}$:

$$\text{RLF} = \frac{1}{n}\sum_{t=1}^{n} L_t(\widetilde{\mathbf{a}}_t, \widetilde{\mathbf{V}}^t) \tag{5.13}$$

It is important to note that, the three settings coincide with the BLPCA and SLPCA problem in Equation (5.6) and (5.10), except the regularization term. However, because of the term $\frac{1}{n}$, the regularization term will be diminishing as n increases. Therefore, the three settings can be used as the evaluation of the LPCA algorithm.

Moreover, RLF is of more interests since it can *sequentially* accumulate the loss functions without waiting til we calculate the last update $\mathbf{V}^n$.

Convergence Analysis

Lemma 5.3.1. *For $t = 1, \cdots, n$ and $L_t(\cdot)$ defined in* (5.8), $\|\nabla_{\mathbf{V}} L_t\|_F \leq \|\mathbf{a}\|_F$, *and* $\|\nabla^2_{\mathbf{V}} L_t\|_{opt} \leq \frac{1}{4}\|\mathbf{a}\|^2_F$.

Proof. W.l.o.g., let $\text{rank}(\mathbf{\Theta}) = 1$, we have:

$$[\nabla_{\mathbf{V}} L_t]_j = -\frac{x^*_{tj}\mathbf{a}_t}{1 + \exp(x^*_{tj}\mathbf{a}_t\mathbf{v}_j^T)}$$

$$\left[\nabla^2_{\mathbf{V}} L_t\right]_{ij} = \left(\frac{x^*_{tj}\mathbf{a}_t\delta_{ij}}{2\cosh(\frac{1}{2}x^*_{tj}\mathbf{a}_t\mathbf{v}_j^T)}\right)^2$$

where $\delta_{ij} = 1$ only when $i = j$ means matrix $\nabla^2_{\mathbf{V}} L_t$ is diagonal. Since $\cosh(x) \geq 1$, hence the norms satisfy $\|\nabla_{\mathbf{V}} L_t\|_F \leq \|\mathbf{a}\|_F$, and $\|\nabla^2_{\mathbf{V}} L_t\|_{opt} \leq \frac{1}{4}\|\mathbf{a}\|^2_F$. □

Lemma 5.3.2. *Let $\widetilde{\mathbf{a}}_t$ be bounded by Ω, for $\forall t = 1, \cdots, n$. Based on* (5.13) *we have* $\|\widetilde{\mathbf{V}}^t - \widetilde{\mathbf{V}}^{t-1}\|_F \leq \eta_t\Omega$.

Proof. From Equation (5.12), we have $\|\widetilde{\mathbf{V}}^t - \widetilde{\mathbf{V}}^{t-1}\|_F = \eta_t\|\nabla_{\mathbf{V}} L_t\|_F$. Since $\widetilde{\mathbf{a}}_t$ result from a regularized problem in (5.10), so $\widetilde{\mathbf{a}}_t$ is bounded by Ω. Thus we have $\|\widetilde{\mathbf{V}}^t - \widetilde{\mathbf{V}}^{t-1}\|_F \leq \eta_t\|\widetilde{\mathbf{a}}_t\|_F \leq \eta_t\Omega$. □

Lemma 5.3.3. *For $L_t(\cdot)$ in* (5.8), $\langle \mathbf{a}, \nabla_{\mathbf{a}} L_t\rangle = \langle \mathbf{V}, \nabla_{\mathbf{V}} L_t\rangle$. *Hence, for $t = 1, \cdots, n$,* $\eta_t\gamma\|\widetilde{\mathbf{a}}_t\|^2_F = \langle \widetilde{\mathbf{V}}^{t-1}, -\eta_t\nabla_{\mathbf{V}} L_t\rangle = \langle \widetilde{\mathbf{V}}^{t-1}, \widetilde{\mathbf{V}}^t - \widetilde{\mathbf{V}}^{t-1}\rangle$.

This follows directly from (5.5) and (5.10).

Lemma 5.3.4. *$L_t(\cdot)$ and surrogate function $\widetilde{L}_t(\cdot)$, as well as their first derivative $\nabla L_t(\cdot)$ and $\nabla \widetilde{h}_t(\cdot)$ are all Lipschitz continuous.*

This is indicated directly from Lemma (5.3.1) & Lemma (5.3.2) and the definition of Lipschitz continuous [6].

Lemma 5.3.5. *For $t = 1, \cdots, n$, if Ω is the upper bound of $\|\mathbf{a}\|^2_{opt}$ as in Lemma (5.3.2), $\|\widetilde{\mathbf{V}}^t\|^2_F \leq \Omega^2 \sum_{s=1}^t \eta_s^2 + 2\gamma\Omega^2 \sum_{s=1}^t \eta_s$.*

Proof. We start from the relationship:

$$\begin{aligned}\|\widetilde{\mathbf{V}}^t - \widetilde{\mathbf{V}}^{t-1}\|^2_F &= \|\widetilde{\mathbf{V}}^t\|^2_F - \|\widetilde{\mathbf{V}}^{t-1}\|^2_F - 2\langle \widetilde{\mathbf{V}}^t - \widetilde{\mathbf{V}}^{t-1}, \widetilde{\mathbf{V}}^{t-1}\rangle \\ &= \|\widetilde{\mathbf{V}}^t\|^2_F - \|\widetilde{\mathbf{V}}^{t-1}\|^2_F - 2\eta_t\gamma\|\widetilde{\mathbf{a}}^t\|^2_F\end{aligned}$$

We sum over the LHS and RHS and get:

$$\sum_{s=1}^t \|\widetilde{\mathbf{V}}^s - \widetilde{\mathbf{V}}^{s-1}\|^2_F + 2\gamma \sum_{s=1}^t \eta_s \|\widetilde{\mathbf{a}}_s\|^2_F = \|\widetilde{\mathbf{V}}^t\|^2_F - \|\widetilde{\mathbf{V}}^0\|^2_F$$

For simplicity, assume $\|\widetilde{\mathbf{V}}^0\|^2_F \approx 0$, we proved the lemma. □

Theorem 5.3.1 (Proposition 2, [54])**.** *Under the regularity condition of Lemma (5.3.4), and $L_t(\cdot)$ a marginally convex function, SLF converges a.s. to BLF.*

The Proof has been implemented in [52] and [54], following a quasi-martingale theory, and use the Bregman divergence under surrogate function as a bridge $\widetilde{L}_t(\cdot)$.

Theorem 5.3.2. *Given step size as $\eta_t = C \times t^{-1/2}$ or $\eta_t = C$, the Regret Loss Function RLF $= \frac{1}{n}\sum_{t=1}^n L_t(\widetilde{\mathbf{a}}_t, \widetilde{\mathbf{V}}^t)$ converges to within a constant to Sequential Loss Function SLF $= \frac{1}{n}\sum_{t=1}^n L_t(\widetilde{\mathbf{a}}_t, \widetilde{\mathbf{V}}^n)$, and thus converges to within a constant of BLF $= \frac{1}{n}\sum_{t=1}^n L_t(\mathbf{a}_t^*, \mathbf{V}^*)$.*

Proof. Based on (5.10) we have:

$$\begin{aligned}\|\widetilde{\mathbf{V}}^t - \widetilde{\mathbf{V}}^n\|^2_F = \|\widetilde{\mathbf{V}}^{t-1} - \widetilde{\mathbf{V}}^n\|^2_F + \eta_t^2\|\nabla_{\mathbf{V}} L_t\|^2_F \\ - 2\eta_t\langle \nabla_{\mathbf{V}} L_t, \widetilde{\mathbf{V}}^{t-1} - \widetilde{\mathbf{V}}^n\rangle\end{aligned}$$

From Lemma (5.3.1), Lemma (5.3.2), and $\|\nabla_{\mathbf{V}} L_t\|_F^2 \leq \Omega^2$, thus:

$$\begin{aligned} n\{RLF - SLF\} &\leq \sum_{t=1}^{n} \langle \nabla_{\mathbf{V}} L_t, \widetilde{\mathbf{V}}^{t-1} - \widetilde{\mathbf{V}}^{n} \rangle \\ &\leq \frac{\|\widetilde{\mathbf{V}}^{n}\|_F^2}{2\eta_0} + \sum_{t=1}^{n} \left(\frac{1}{2\eta_t} - \frac{1}{2\eta_{t-1}} \right) \|\widetilde{\mathbf{V}}^{n} - \widetilde{\mathbf{V}}^{t-1}\|_F^2 + \frac{\Omega^2}{2}\eta_t \\ &\leq \frac{\|\widetilde{\mathbf{V}}^{n}\|_F^2}{2\eta_0} + \sum_{t=1}^{n} \left(\frac{1}{2\eta_t} - \frac{1}{2\eta_{t-1}} \right) \|\widetilde{\mathbf{V}}^{n}\|_F^2 + \frac{\Omega^2}{2}\eta_t \end{aligned}$$

- diminishing step size $\eta_t = Ct^{-1/2}$. From Lemma (5.3.5), we have:

$$\begin{aligned} |RLF - SLF| \leq \frac{\Omega^2 C}{2}\frac{\log n}{n} + \frac{\Omega^2 C}{4}\frac{\log n}{\sqrt{n}} \\ + \frac{\Omega^2(2\gamma + C)}{2\sqrt{n}} + \frac{\gamma\Omega^2}{2} \end{aligned}$$

  Then $\lim_{n\to\infty} |RLF - SLF| \leq \frac{\gamma\Omega^2}{2}$. But with reasonable n, the term $\frac{\Omega^2 C \log n}{\sqrt{n}}$ will also be significant. Usually, small C and γ can force a lower error bound. However, small γ can result in more steps in optimizing for $\widetilde{\mathbf{a}}_t$, whereas small C would make the step size too small, which may not be a good choice if we want a fast decaying of the error bound.

- constant step size $\eta_t = C$: For constant step, we have:

$$|RLF - SLF| \leq \gamma\Omega^2 + \Omega^2 C$$

  Similarly, we prefer small small C and γ. The challenge of using small C and γ has already been discussed.

□

Principal Component Selection Criterion

Conventional PCA evaluates Principal Component (PC) selection by the amount of variance the PCs capture. In LPCA, this is not working, and we need to find other criterion.

Since we are maximizing the likelihood function, an intuitive way is to evaluate the likelihood as below:

$$L = \sum_{i,j} \log \left[g(\theta_{ij})^{X_{ij}} \left(1 - g(\theta_{ij})\right)^{1 - X_{ij}} \right] \tag{5.14}$$

in which $g(\theta) = \left(1 + e^{-\theta}\right)^{-1}$.

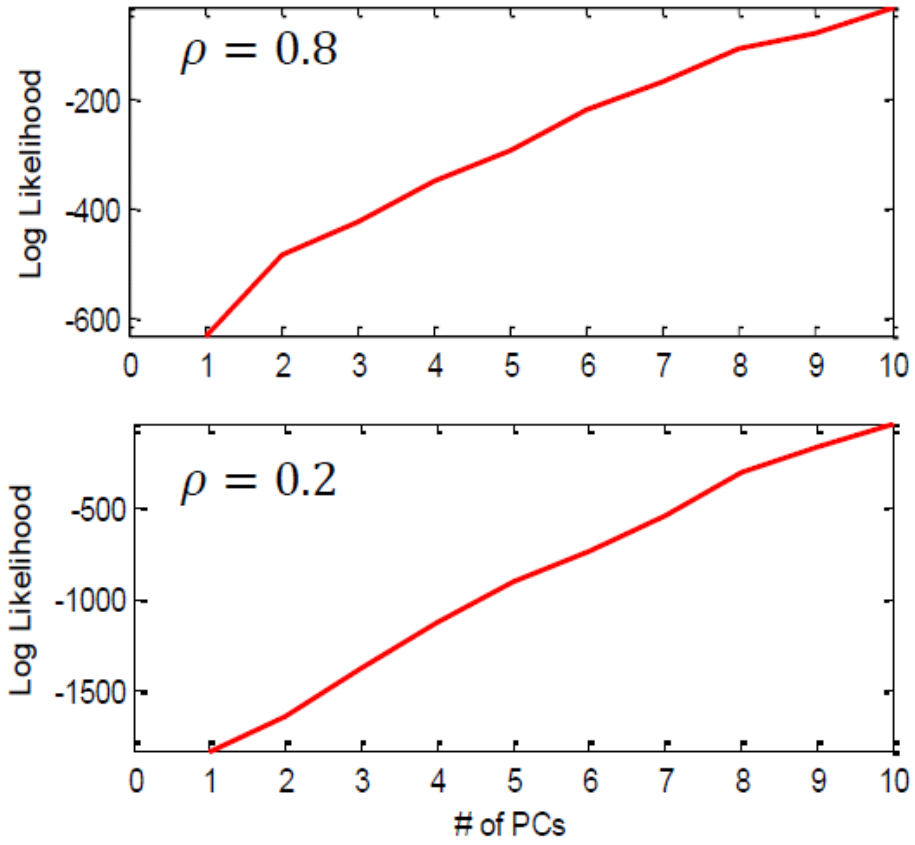

Figure 5.1: Log Likelihood as a function of the number of Principal Components taken, based on simulated correlated 10-dimensional binary sequences, with correlation factor equals to 0.8 (upper) and 0.2 (lower).

We can show this from Figure 5.1, which shows the log likelihood as a function of the number of principal components under different correlation factors.

However, in most cases likelihood function is hard to evaluate. Another intuitive way is to study the accuracy the PCs carry. As we recover the original data, we cannot recover the $0, 1$ multivariate data. Instead, we recover the natural parameter $g(\theta) = \left(1 + e^{-\theta}\right)^{-1}$, which is a real number between 0 and 1. If the recovery is 0.7, then there is 70% chance that we will recover the right state. For multivariate data, we can calculate the *average* error rate. The error rate function is:

$$Err = 1 - \sum_{i,j} \left[g(\theta_{ij})^{X_{ij}} \left(1 - g(\theta_{ij})\right)^{1 - X_{ij}} \right] \tag{5.15}$$

Following the Jensen's equality, the accuracy function Err is roughly an upper bound of the likelihood function. We show the result in Figure 5.2. As expected, for highly-correlated ($\rho = 0.8$) sequences, first principal component is enough to capture roughly 90% accuracy, whereas for $\rho = 0.2$, first principal component can only capture 60% accuracy.

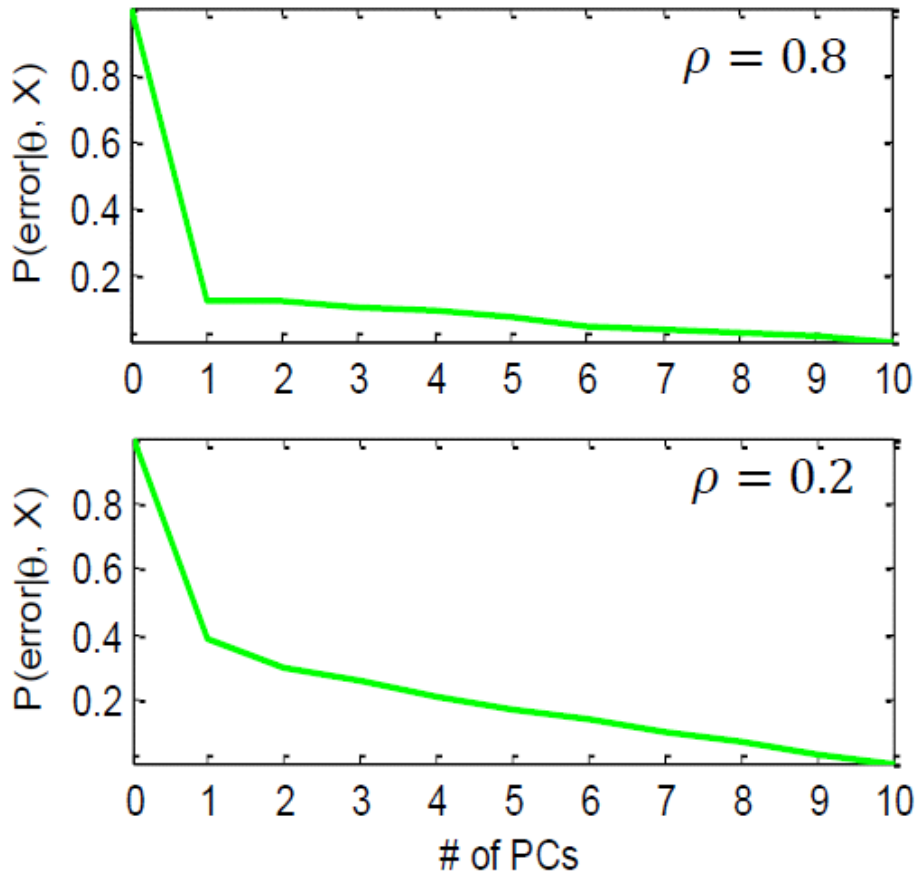

Figure 5.2: Average accuracy as a function of the number of Principal Components taken, based on simulated correlated 10-dimensional binary sequences, with correlation factor equals to 0.8 (upper) and 0.2 (lower).

5.4 Experimental Results

Simulated Binary-State System

Firstly, simulated binary data was used to test the performance of the SLPCA algorithm in a binary-state system. The generation of correlated Bernoulli sequences is illustrated in [50]. This work focused on the case where $\text{rank}(\mathbf{\Theta}) = 1$, since this usually demonstrates the best dimension reduction capability. It should be noted here that the extension to multiple principal components is straightforward, following the iterative updating rules in [12].

We tried the above on data with $P = 8$ dimension and length of $n = 1000$ data points. We initialize $\widetilde{\mathbf{V}}^0$ such that its norm is close but not equal to zero, for computation and convergence purposes. Fig 5.3 shows the three functions defined in (5.11) to (5.13); whereas Fig 5.4 shows the key parameters in the sequential steps. There are some interesting findings.

Firstly, though both SLF and RLF converges at least within a constant to BLF, the stochastic learning can be clearly divided into three Phases, as shown in Fig 5.3. Phase I stands for the period when the norm of $\widetilde{\mathbf{V}}^0$ is close to zero right after the initialization, when $L_t(\mathbf{a}_t, \mathbf{V})$ approaches $P \log 2$ as in Equation (5.8). Phase II characterizes the decay of error versus n, whereas Phase III stands for when the error converges to within a constant independent of n.

Secondly, $\|\widetilde{\mathbf{V}}^t\|_F^2$ increases versus t, which means that $\|\widetilde{\mathbf{V}}^t\|_F^2$ behaves differently from the coefficient in sequential learning of linear model [52] [54]. Matrix factorization places no constraints for $\widetilde{\mathbf{V}}^t$, hence cannot guarantee the bound of $\widetilde{\mathbf{V}}^t$. From another perspective, $\widetilde{\mathbf{a}}_t$ is bounded since Equation (5.10) has fixed in size, while $\widetilde{\mathbf{V}}_t$ not since there is a summation of loss functions. It should be noted that, in Fig 5.4, $\widetilde{\mathbf{a}}_t$ decreases versus t, which could result from (5.9) and is an interesting topic in the future.

Thirdly, due to the unbounded $\widetilde{\mathbf{V}}^t$, the term $\|\widetilde{\mathbf{V}}^t - \widetilde{\mathbf{V}}^{t-1}\|_F$ is not $\propto t^{-1}$ as in [52] and [54]. It should be noted that the theoretical bound for $\|\widetilde{\mathbf{V}}^t - \widetilde{\mathbf{V}}^{t-1}\|_F$ under constant step size could be as low as $t^{-1/2}$, which could be a result of the convergence behavior of $\widetilde{\mathbf{a}}_t$ under constant step size.

Last but not least, it is important to mention that the bounds obtained in Theorem (5.3.2) assume n large enough. However, in many cases the decay of n is not that fast. Therefore, the effect of n cannot be completely ignored in the analysis.

Building End-Use Energy Modeling

As illustrated before, *Top-Down* monitoring of building end-use is usually implemented as a statistical filter. For *Bottom-Up* monitoring, however, we need to track multiple-dimensional occupant-behavioral sequences. SLPCA is able to extract a *Principal Appliance* out of the multivariate sequences to characterize the whole space occupant behavior.

As an example field study, we focus on *Bottom-Up* monitoring of the multiple-computer-monitors system. The computer monitors are located in CREST space at University of California, Berkeley. We collect the power consumption of 6 monitors in 10 minutes interval by ACme sensor network[3] through CoreSight OsiSoft system[4]. We take five days data, which is roughly 720 data points. The real power sequences are filtered into ON/OFF states by power disaggregation algorithm [37]. The individual as well as the aggregated ON/OFF state sequences are shown in Figure 5.6. With the ON/OFF states, we then use BLPCA and SLPCA to obtain the *Principal Appliance* of the building.

In our SLPCA, we choose constant step size that is short enough to track the changes as they appear[5]. We also only consider the first *Principal Appliance* since more than 90% accuracy can be achieved. The convergence of the algorithm is shown in Figure 5.5. We observe a good convergence for both SLF and RLF. Periodic fluctuation is observed, due to the periodic transition between day and night energy consumption, which results in periodical changing of the data model. Moreover, the online algorithm demonstrates less fluctuation because they adaptively update the model of the data.

We reconstruct the original data with three sets of variables: the BLF setting $\mathbf{A}^*$, $\mathbf{V}^*$; the SLF setting $\{\widetilde{\mathbf{a}}_t\}$, $\widetilde{\mathbf{V}}^n$; and the RLF setting $\{\widetilde{\mathbf{a}}_t\}$, $\{\widetilde{\mathbf{V}}^t\}$. The results are compared with the original data in Figure 5.7 (sum of states of all appliances, 1 as ON and 0 as OFF). Interestingly, SLF setting gives better approximation to BLF setting since it is more adaptive in terms of $\widetilde{\mathbf{V}}^t$ and can better catch the periodic pattern of the original data. On the other hand, BLF setting uses the $\widetilde{\mathbf{V}}^n$, which could give unpromising result if data is non-stationary.

[3] http://acme.cs.berkeley.edu/

[4] http://picoresight.osisoft.com/

[5] one could presumably also leverage the likely periodic behavior of the data by appropriate aggregation

5.5 Conclusions and Future Tasks

In this Chapter, dimensionality reduction in *Bottom-Up* end-use monitoring is discussed. A logistic PCA (LPCA) is applied to accommodate the traditional PCA to the multivariate binary data in *Bottom-Up* end-use setting. To adapt the LPCA to streaming data and fast online application, a sequential version of LPCA (SLPCA) was developed based on online convex optimization theory, which can achieve computational and storage efficiency. In this study, two functions to evaluate the SLPCA algorithm were defined (i.e., the Sequential Loss Function, or SLF and the Regret Loss Function, or RLF), and it was shown that both of them converge at least within a constant to offline batch LPCA (BLPCA) results. An application of this algorithm in building end-use monitoring was eventually demonstrated.

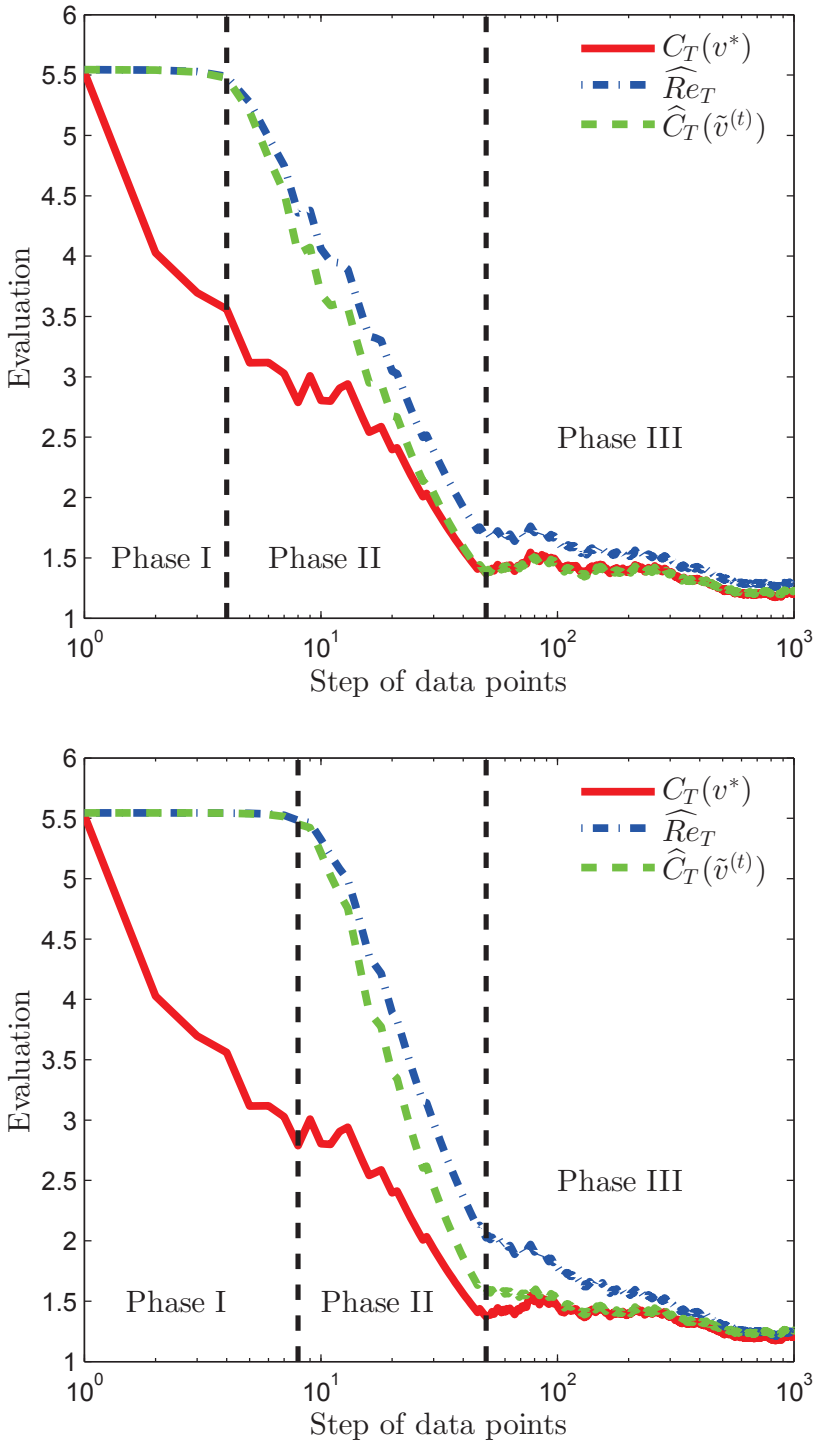

Figure 5.3: The three functions BLF, SLF and RLF as function of t. Top: $\eta_t = Ct^{-1/2}$, with $C = 0.2$, $\gamma = 0.1$. Bottom: $\eta_t = C$, with $C = 0.05$, $\gamma = 0.1$.

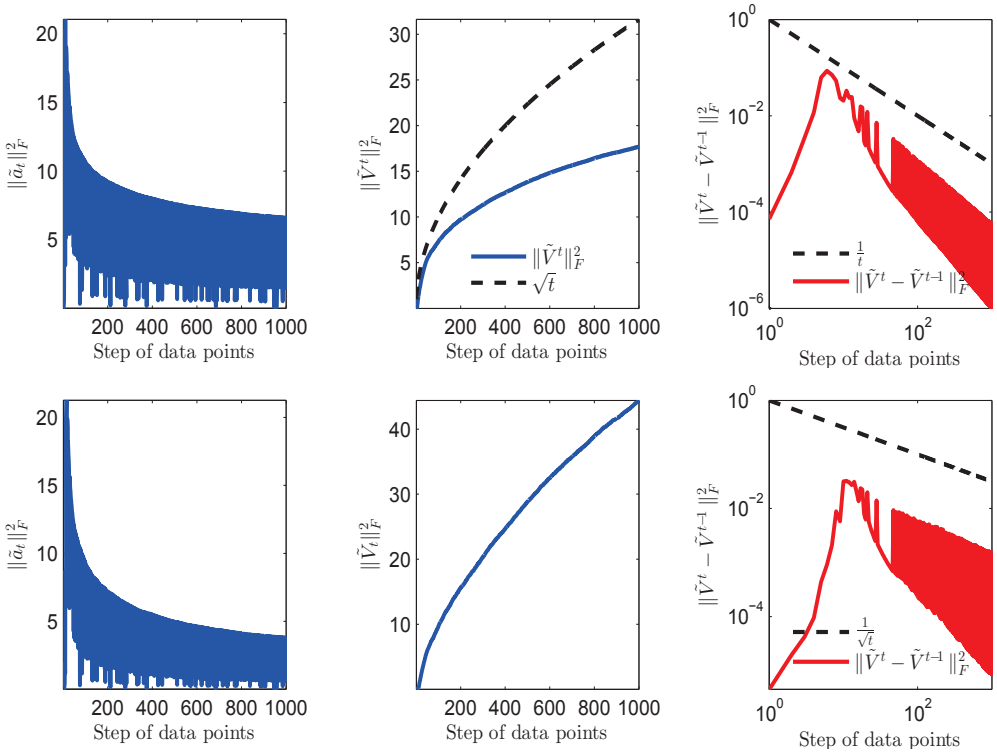

Figure 5.4: The convergence property of $\widetilde{\mathbf{a}}_t$, $\widetilde{\mathbf{V}}^t$ and $\|\widetilde{\mathbf{V}}^t - \widetilde{\mathbf{V}}^{t-1}\|_F$. Top: $\eta_t = Ct^{-1/2}$, with $C = 0.2$, $\gamma = 0.1$. Bottom: $\eta_t = C$, with $C = 0.05$, $\gamma = 0.1$.

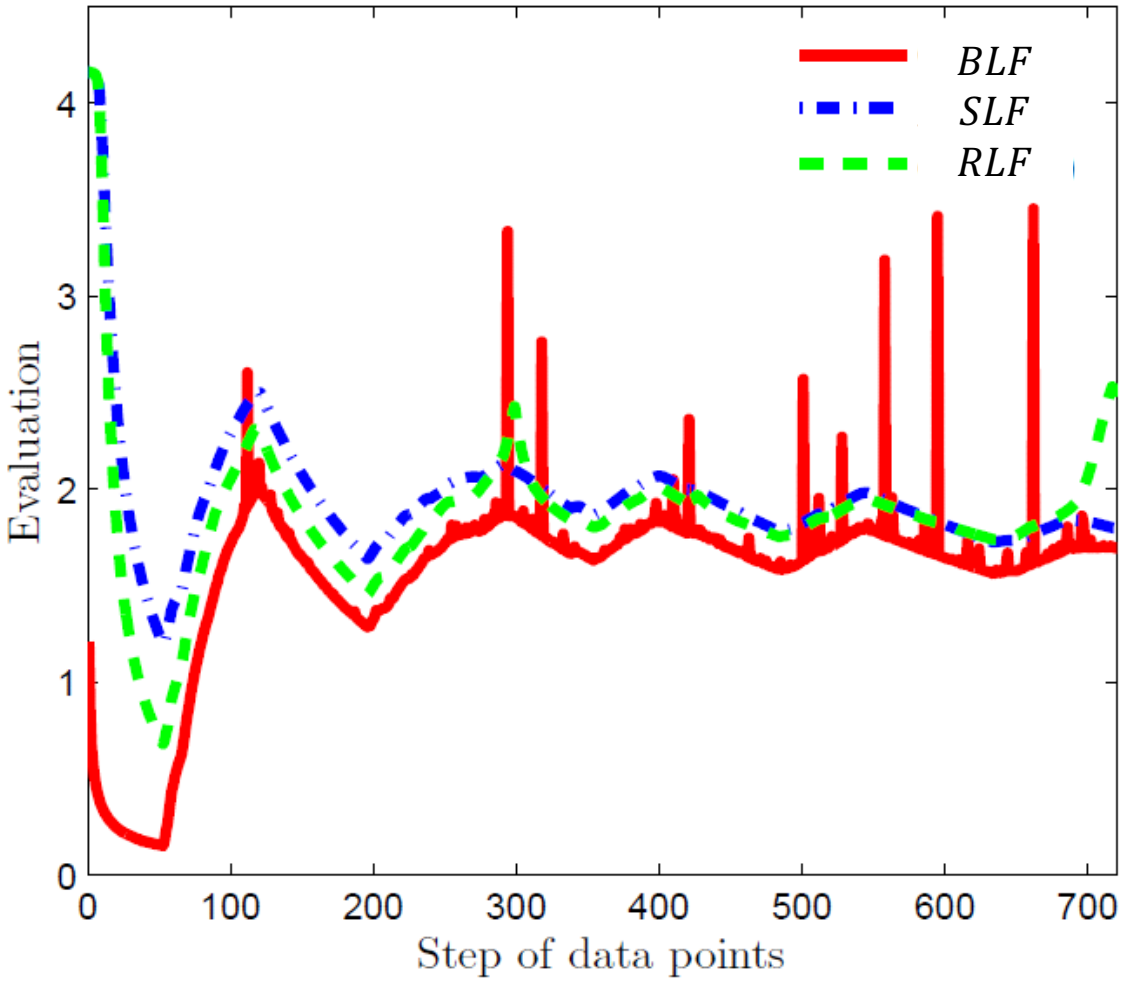

Figure 5.5: The three functions BLF, SLF and RLF as function of t for energy end-use simulation with constant step size $\eta_t = C$ as $C = 0.05$, $\gamma = 0.1$.

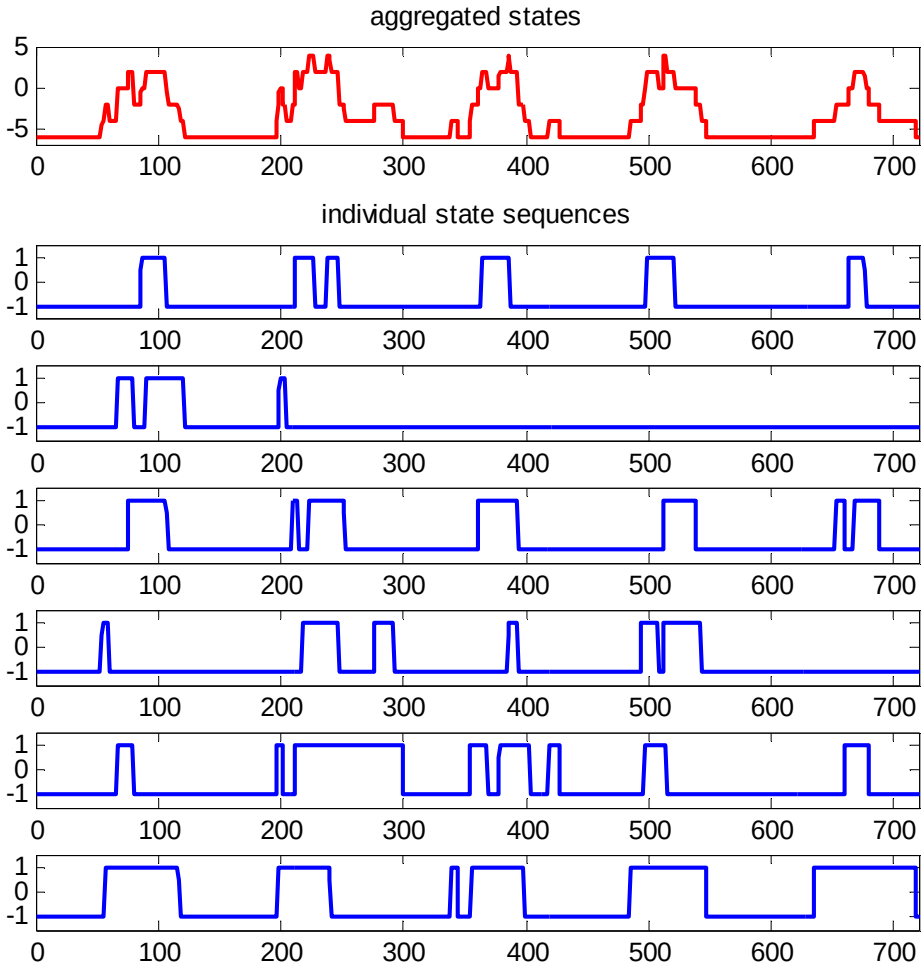

Figure 5.6: The individual as well as the aggregated ON/OFF sequences of six computer monitors.

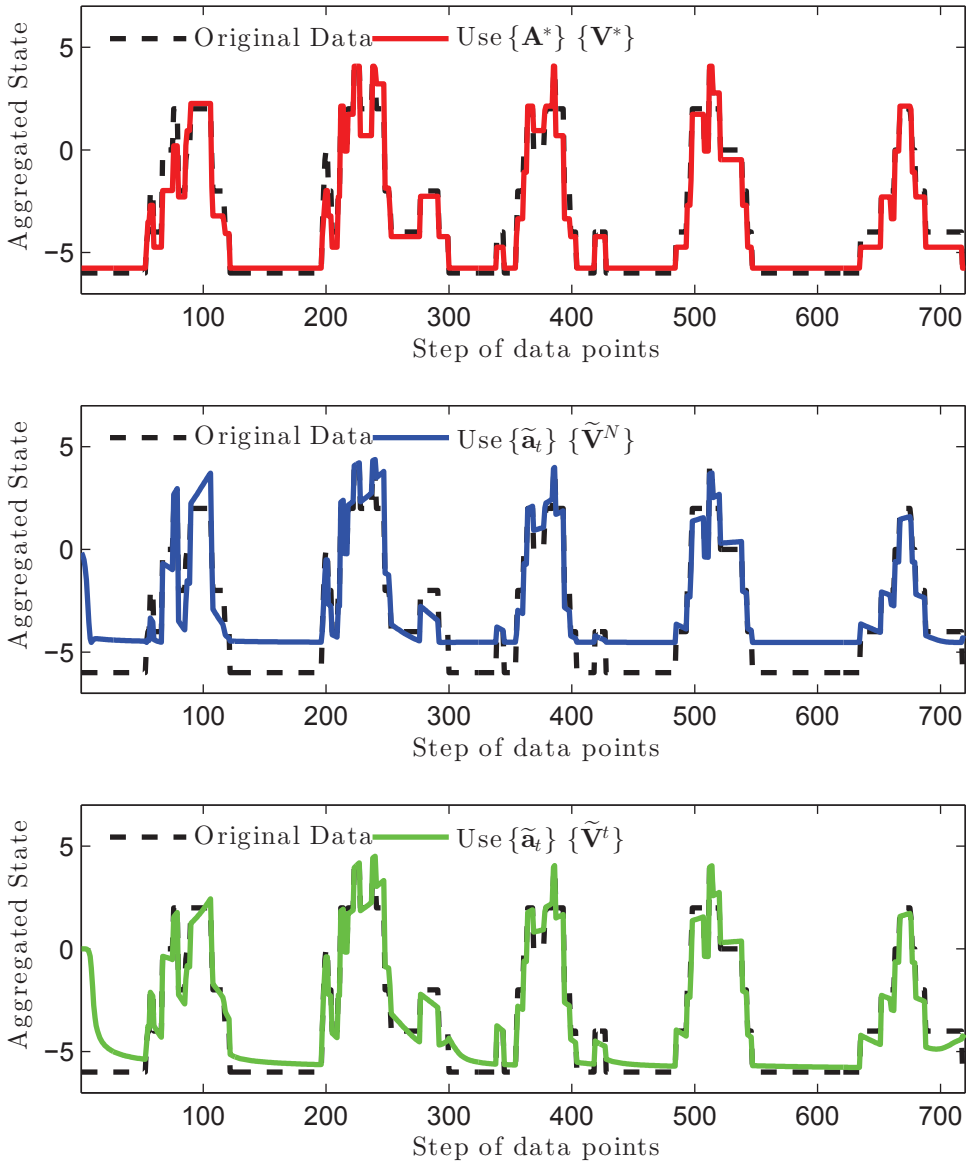

Figure 5.7: Reconstruction of the aggregated state (sum of states of 6 monitors) under the three sets of variables.

Chapter 6

Conclusion and Future Tasks

In this work, the *modeling* and *monitoring* of the end-use of commercial buildings are studied. Two types of the most widely used methods, *Top-Down* approaches and *Bottom-Up* approaches, were investigated and compared while current issues were addressed.

In the *Top-Down* approach, an ASVR model was developed to accommodate the nonlinearity and nonstationarity of the macro-level time series that is difficult to solve in a linear autoregressive model. A future task in this work would be to design the change recognition function to deal with new non-ideal patterns, especially in monitoring and fault diagnosis application.

In the *Bottom-Up* approach, an appliance-data-driven stochastic model based on ON/OFF switching events was built to estimate the power consumption of each end-use sector of a commercial building. Future tasks include a better modeling of shared appliances and a more reasonable modeling of inter-appliance correlation.

Power disaggregation techniques used in *Bottom-Up* end-use *monitoring* and *modeling* were also discussed. Conventional methods of power disaggregation, including HMM and Edge-Driven models were studied and compared, with new methods based on multi-hypothesis sequential testing algorithm proposed to overcome impulse noise. With power disaggregation technique to obtain appliance ON/OFF states, the appliance-data-driven *Bottom-Up* model was demonstrated in real commercial buildings under different scenarios, along with its capability to estimate the end-use power consumption of commercial buildings.

Finally, *monitoring* in *Bottom-Up* settings was studied. Dimensionality reduction technique was applied to achieve efficient monitoring; in order to accommodate to the streaming multivariate binary-state occupant-behavioral data, logistic PCA (LPCA) was chosen as a tool and extended to a sequential version, as SLPCA. In the future, it is needed to further improve the convergence and performance of SLPCA through a

more efficient online convex optimization algorithm. A more intuitive way to quantify dimensionality reduction in binary data is also needed.

Bibliography

[1] Jacob Abernethy et al. "A new approach to collaborative filtering: Operator estimation with spectral regularization". In: *The Journal of Machine Learning Research* 10 (2009), pp. 803–826.

[2] Dennis J Aigner, Cyrus Sorooshian, and Pamela Kerwin. "Conditional demand analysis for estimating residential end-use load profiles". In: *The Energy Journal* (1984), pp. 81–97.

[3] Arindam Banerjee et al. "Clustering with Bregman divergences". In: *The Journal of Machine Learning Research* 6 (2005), pp. 1705–1749.

[4] M. Basseville and I.V. Nikiforov. *Detection of abrupt changes: theory and application*. Vol. 104. Prentice Hall Englewood Cliffs, NJ, 1993.

[5] C.W. Baum and V.V. Veeravalli. "A sequential procedure for multihypothesis testing". In: *Information Theory, IEEE Transactions on* 40.6 (1994).

[6] Dimitri P Bertsekas. "Nonlinear programming". In: (1999).

[7] P.J. Bickel and K.A. Doksum. *Mathematical Statistics, volume I*. 2001.

[8] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2009.

[9] Lev M Bregman. "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming". In: *USSR computational mathematics and mathematical physics* 7.3 (1967), pp. 200–217.

[10] A Capasso et al. "A bottom-up approach to residential load modeling". In: *Power Systems, IEEE Transactions on* 9.2 (1994), pp. 957–964.

[11] Steven Chu and Arun Majumdar. "Opportunities and challenges for a sustainable energy future". In: *nature* 488.7411 (2012), pp. 294–303.

[12] Michael Collins, Sanjoy Dasgupta, and Robert E Schapire. "A generalization of principal components analysis to the exponential family". In: *Advances in neural information processing systems*. 2001, pp. 617–624.

[13] Corinna Cortes and Vladimir Vapnik. "Support-vector networks". In: *Machine learning* 20.3 (1995), pp. 273–297.

[14] Drury B Crawley et al. "Contrasting the capabilities of building energy performance simulation programs". In: *Building and environment* 43.4 (2008), pp. 661–673.

[15] I Csisz, Gábor Tusnády, et al. "Information geometry and alternating minimization procedures". In: *Statistics and decisions* (1984).

[16] Stephen Dawson-Haggerty et al. "sMAP: a simple measurement and actuation profile for physical information". In: *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*. ACM. 2010, pp. 197–210.

[17] *DENT meter*. http://www.dentinstruments.com/. Accessed: 2014-04-10.

[18] US DOE. "Building energy software tools directory". In: *Washington, DC. Accessed November* 27 (2012), p. 2012.

[19] US DOE. "Quadrennial Technology Review 2011". In: *US Department of Energy, Washington, DC* (2011).

[20] M. Dong et al. "An Event Window Based Load Monitoring Technique for Smart Meters". In: *Smart Grid, IEEE Transactions on* 3.2 (2012), pp. 787–796.

[21] J. Duan et al. "Neural network approach for estimation of load composition". In: *Circuits and Systems, 2004. ISCAS'04. Proceedings of the 2004 International Symposium on*. Vol. 5. IEEE. 2004, pp. V–988.

[22] M.R. Durling et al. *Cognitive electric power meter*. EP Patent 2,026,299. Feb. 2009.

[23] US EIA. "Annual energy outlook 2014". In: *US Energy Information Administration, Washington, DC* (2014).

[24] Yaakov Engel, Shie Mannor, and Ron Meir. "The kernel recursive least-squares algorithm". In: *Signal Processing, IEEE Transactions on* 52.8 (2004), pp. 2275–2285.

[25] Y. Ephraim and W.J.J. Roberts. "Revisiting autoregressive hidden Markov modeling of speech signals". In: *Signal Processing Letters, IEEE* 12.2 (2005), pp. 166–169.

[26] Jiashi Feng, Huan Xu, and Shuicheng Yan. "Online robust PCA via stochastic optimization". In: *Advances in Neural Information Processing Systems*. 2013, pp. 404–412.

[27] E.B. Fox et al. "An HDP-HMM for systems with state persistence". In: *Proc. International Conference on Machine Learning*. Vol. 2. IEEE Press Piscataway, NJ. 2008.

[28] Z. Ghahramani and M.I. Jordan. "Factorial hidden Markov models". In: *Machine learning* 29.2 (1997), pp. 245–273.

[29] Arnaud Grandjean, Jérôme Adnot, and Guillaume Binet. "A review and an analysis of the residential electric load curve models". In: *Renewable and Sustainable Energy Reviews* 16.9 (2012), pp. 6539–6565.

[30] G.W. Hart. "Nonintrusive appliance load monitoring". In: *Proceedings of the IEEE* 80.12 (1992), pp. 1870–1891.

[31] Trevor Hastie et al. *The elements of statistical learning*. Vol. 2. 1. Springer, 2009.

[32] P.J. Huber. *Robust statistical procedures*. Vol. 68. SIAM, 1996.

[33] Xiaofan Jiang et al. "Design and implementation of a high-fidelity ac metering network". In: *Information Processing in Sensor Networks, 2009. IPSN 2009. International Conference on*. IEEE. 2009, pp. 253–264.

[34] M.J. Johnson and A.S. Willsky. "Bayesian Nonparametric Hidden Semi-Markov Models". In: *arXiv preprint arXiv:1203.1365* (2012).

[35] Soteris A Kalogirou. "Applications of artificial neural-networks for energy systems". In: *Applied Energy* 67.1 (2000), pp. 17–35.

[36] Zhaoyi Kang, Ming Jin, and Costas J Spanos. "Modeling of End-Use Energy Profile: An Appliance-Data-Driven Stochastic Approach". In: *arXiv preprint arXiv:1406.6133* (2014).

[37] Zhaoyi Kang et al. "Virtual power sensing based on a multiple-hypothesis sequential test". In: *Smart Grid Communications (SmartGridComm), 2013 IEEE International Conference on*. IEEE. 2013, pp. 785–790.

[38] Holger Karl and Andreas Willig. *Protocols and architectures for wireless sensor networks*. Wiley-Interscience, 2007.

[39] Wolfgang Kastner et al. "Communication systems for building automation and control". In: *Proceedings of the IEEE* 93.6 (2005), pp. 1178–1203.

[40] Sila Kiliccote, Mary Ann Piette, and David Hansen. "Advanced controls and communications for demand response and energy efficiency in commercial buildings". In: *Lawrence Berkeley National Laboratory* (2006).

[41] H.S. Kim. "Unsupervised disaggregation of low frequency power measurements". PhD thesis. University of Illinois, 2012.

[42] Jyrki Kivinen, Alexander J Smola, and Robert C Williamson. "Online learning with kernels". In: *Signal Processing, IEEE Transactions on* 52.8 (2004), pp. 2165–2176.

[43] J.Z. Kolter and T. Jaakkola. "Approximate Inference in Additive Factorial HMMs with Application to Energy Disaggregation". In: *International Conference on Artificial Intelligence and Statistics*. 2012.

[44] J.Z. Kolter and M.J. Johnson. "REDD: A public data set for energy disaggregation research". In: *Workshop on Data Mining Applications in Sustainability (SIGKDD), San Diego, CA*. 2011.

[45] Steven Lanzisera et al. "Data network equipment energy use and savings potential in buildings". In: *Energy Efficiency* 5.2 (2012), pp. 149–162.

[46] Fei Lei and Pingfang Hu. "A baseline model for office building energy consumption in hot summer and cold winter region". In: *Management and Service Science, 2009. MASS'09. International Conference on*. IEEE. 2009, pp. 1–4.

[47] Qiong Li et al. "Applying support vector machine to predict hourly cooling load in the building". In: *Applied Energy* 86.10 (2009), pp. 2249–2256.

[48] J. Liang et al. "Load signature study-Part I: Basic concept, structure, and methodology". In: *Power Delivery, IEEE Transactions on* 25.2 (2010), pp. 551–560.

[49] Weifeng Liu, Puskal P Pokharel, and Jose C Principe. "The kernel least-mean-square algorithm". In: *Signal Processing, IEEE Transactions on* 56.2 (2008), pp. 543–554.

[50] A Daniel Lunn and Stephen J Davies. "A note on generating correlated binary variables". In: *Biometrika* 85.2 (1998), pp. 487–490.

[51] Junshui Ma, James Theiler, and Simon Perkins. "Accurate on-line support vector regression". In: *Neural Computation* 15.11 (2003), pp. 2683–2703.

[52] Julien Mairal et al. "Online learning for matrix factorization and sparse coding". In: *The Journal of Machine Learning Research* 11 (2010), pp. 19–60.

[53] Morteza Mardani, Gonzalo Mateos, and G Giannakis. "Decentralized Sparsity-Regularized Rank Minimization: Algorithms and Applications". In: *Signal Processing, IEEE Transactions on* 61.21 (2013), pp. 5374–5388.

[54] Morteza Mardani, Gonzalo Mateos, and Georgios B Giannakis. "Rank minimization for subspace tracking from incomplete data". In: *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE. 2013, pp. 5681–5685.

[55] Johanna L Mathieu et al. "Quantifying changes in building electricity use, with application to demand response". In: *Smart Grid, IEEE Transactions on* 2.3 (2011), pp. 507–518.

[56] Naoya Motegi et al. "Introduction to commercial building control strategies and techniques for demand response". In: *Lawrence Berkeley National Laboratory LBNL-59975* (2007).

[57] Peter Palensky and Dietmar Dietrich. "Demand side management: Demand response, intelligent energy systems, and smart loads". In: *Industrial Informatics, IEEE Transactions on* 7.3 (2011), pp. 381–388.

[58] Spiros Papadimitriou, Jimeng Sun, and Christos Faloutsos. "Streaming pattern discovery in multiple time-series". In: *Proceedings of the 31st international conference on Very large data bases*. VLDB Endowment. 2005, pp. 697–708.

[59] O. Parson et al. "Nonintrusive load monitoring using prior models of general appliance types". In: *26th AAAI Conference on Artificial Intelligence*. 2012.

[60] H.V. Poor. "An introduction to signal detection and estimation". In: *New York, Springer-Verlag, 1988, 559 p.* 1 (1988).

[61] Cédric Richard, José Carlos M Bermudez, and Paul Honeine. "Online prediction of time series data with kernels". In: *Signal Processing, IEEE Transactions on* 57.3 (2009), pp. 1058–1067.

[62] Ian Richardson et al. "Domestic electricity use: A high-resolution energy demand model". In: *Energy and Buildings* 42.10 (2010), pp. 1878–1887.

[63] Sheldon M Ross. *Introduction to probability models*. Academic press, 2006.

[64] Shai Shalev-Shwartz. "Online learning and online convex optimization". In: *Foundations and Trends in Machine Learning* 4.2 (2011), pp. 107–194.

[65] Alex J Smola and Bernhard Schölkopf. "A tutorial on support vector regression". In: *Statistics and computing* 14.3 (2004), pp. 199–222.

[66] Rajesh Subbiah et al. "A high resolution energy demand model for commercial buildings". In: *Security in Critical Infrastructures Today, Proceedings of International ETG-Congress 2013; Symposium 1:* VDE. 2013, pp. 1–6.

[67] Lukas G Swan and V Ismet Ugursal. "Modeling of end-use energy consumption in the residential sector: A review of modeling techniques". In: *Renewable and Sustainable Energy Reviews* 13.8 (2009), pp. 1819–1835.

[68] Michael E Tipping and Christopher M Bishop. "Probabilistic principal component analysis". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61.3 (1999), pp. 611–622.

[69] Vladimir Naumovich Vapnik and Vlamimir Vapnik. *Statistical learning theory.* Vol. 2. Wiley New York, 1998.

[70] Rene Vidal, Yi Ma, and Shankar Sastry. "Generalized principal component analysis (GPCA)". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 27.12 (2005), pp. 1945–1959.

[71] MP Wand and MC Jones. "Multivariate plug-in bandwidth selection". In: *Computational Statistics* 9.2 (1994), pp. 97–116.

[72] Y. Wang et al. "Tracking states of massive electrical appliances by lightweight metering and sequence decoding". In: *Proceedings of the Sixth International Workshop on Knowledge Discovery from Sensor Data.* ACM. 2012, pp. 34–42.

[73] Thomas Weng et al. "Managing plug-loads for demand response within buildings". In: *Proceedings of the Third ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings.* ACM. 2011, pp. 13–18.

[74] Joakim Widén and Ewa Wäckelgård. "A high-resolution stochastic model of domestic activity patterns and electricity demand". In: *Applied Energy* 87.6 (2010), pp. 1880–1892.

[75] Urs Wilke et al. "A bottom-up stochastic model to predict building occupants' time-dependent activities". In: *Building and Environment* 60 (2013), pp. 254–264.

[76] M. Zeifman and K. Roth. "Nonintrusive appliance load monitoring: Review and outlook". In: *Consumer Electronics, IEEE Transactions on* 57.1 (2011), pp. 76–84.

[77] Fei Zhao. "Agent-based modeling of commercial building stocks for energy policy and demand response analysis". In: (2012).

[78] Hai-xiang Zhao and Frédéric Magoulès. "A review on the prediction of building energy consumption". In: *Renewable and Sustainable Energy Reviews* 16.6 (2012), pp. 3586–3592.

[79] Zhi Zhou, Fei Zhao, and Jianhui Wang. "Agent-based electricity market simulation with demand response from commercial buildings". In: *Smart Grid, IEEE Transactions on* 2.4 (2011), pp. 580–588.

[80] Martin Zinkevich. "Online convex programming and generalized infinitesimal gradient ascent". In: (2003).