
# A Non-Intrusive Load Monitoring Framework for Robust Real-Time Disaggregation of Smart Meter Data

YUNG FEI (RICKY) WONG

SUBMITTED IN TOTAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

*Department of Electrical and Computer Systems Engineering*
FACULTY OF ENGINEERING
MONASH UNIVERSITY
AUSTRALIA

AUGUST 2017

# Copyright Notice

Monash University
Australia

## ABSTRACT

# A Non-Intrusive Load Monitoring Framework for Robust Real-Time Disaggregation of Smart Meter Data

YUNG FEI (RICKY) WONG

Non-intrusive Load Monitoring (NILM) is a class of techniques used for decomposing aggregate electrical measurements into their contributing appliances. It only requires a single sensing point to infer the appliance-specific decomposition of energy consumption in a residential unit. Fundamental to this is to identify and model the associations between the appliances and their induced electrical patterns observable in the aggregate signal.

However, it is not uncommon to encounter situations where the association is not one-to-one; different appliances can induce similar patterns to result in poor disaggregation accuracy – a problem that becomes especially true when only sparsely sampled, scalar-valued aggregate data is available (like from smart meters). In addition, there is the rarely addressed issue related to the presence of appliances for which the aforementioned associations are not yet modelled. If these unknown/unmodelled appliances are not considered in the design of NILM algorithms, the estimation of energy consumption of known appliances can be severely affected. This is on top of the further difficulty in distinguishing between unknown appliances and known appliances with similar patterns, and the computational challenges in performing disaggregation under a more complex but powerful appliance model. To address these challenges, this thesis proposes a new framework for NILM with improved disaggregation accuracy and increased robustness, while maintaining computational efficiency for real-time applications.

First, a new instance of the hidden semi-Markov model for NILM, adapted from the field of acoustic speech modelling, is introduced to capture salient appliance state duration information. The proposed model, named *factorial variable transition hidden Markov model*, enables incremental calculation of time-varying, duration-dependent state transition probabilities, to allow for the separation of appliances with similar patterns. To evaluate its effectiveness in the worst case scenario, synthetic aggregate data consisting of appliances with exactly the same power consumption is generated and disaggregated. It was found that the proposed model is highly successful in reconstructing the appliance-level contributions, as compared to the state-of-the-art methods based on hidden Markov models (HMMs).

Second, a novel method, *Particle-based Distribution Truncation (PBDT)*, for performing state inference under the proposed model is provided. Unlike standard exact methods, which are computationally intractable, and standard approximation algorithms, which are inherently batch-processed, PBDT is a fast real-time approximation algorithm with good convergence properties. This is achieved by combining the dynamic programming paradigm of the Viterbi algorithm and the survival-of-the-fittest principle from particles filters, whereby unlikely solutions are intelligently pruned and computation results amongst groups of related particles are shared. Together with the proposed model, the outcome is a computationally scalable approach for disaggregation. In the evaluation over the data of real houses from the Reference Energy Disaggregation Dataset (REDD), it was discovered that an average sub-second per-sample processing time is achievable for problems with as many as 20 billion states, while achieving a disaggregation accuracy of approximately 80%. Further, empirical results also illustrate near linear growth in time complexity as the number of appliances to be extracted increases, establishing PBDT as a scalable approach even when a more complex and powerful model is used.

Third, the benefits of incorporating slowly-decaying power features, common to certain appliances (e.g. refrigerators), were investigated to further resolve similarities between different appliances. This is achieved by relaxing the constraint that the power consumption for a fixed state is stationary and augmenting the initially proposed model with relationships between the decay in power consumption and the appliance state duration. The outcome is a specific instance of a segmental HMM, like those used in speech pattern modelling but with inherited benefits of the initial model. Results from the evaluation with the REDD dataset indicate an additional improvement of approximately 5% in disaggregation accuracy, with errors due to modelling constraints largely corrected.

Finally, the problem of unmodelled appliances was addressed by extending the initial model via an additional residual term for capturing unknown contributions of power. The term is imposed with a robust noise model based on compressed sensing to penalise patterns due to unknown loads. Also included is a steady-state segmentation algorithm based on an earlier work to guard against spurious spikes in power.

When combined with these additions, the initially proposed model and an extended PBDT algorithm form a robust real-time disaggregation framework for NILM. Experimental results using the REDD dataset show that, for each house, the extraction of the top 5 most energy-consuming appliances amongst unmodelled loads is highly successful, with an average correct energy assignment rate of 83%.

# Declaration

In accordance with Monash University Doctorate Regulation 17.2: *Doctor of Philosophy and Research Master's Regulations*, the following declarations are made:

I hereby declare that this thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

The core theme of the thesis is the creation of a new framework for real-time load disaggregation. The ideas, development and writing up of all the work in the thesis were the principal responsibility of myself, the candidate, working within the Department of Electrical and Computer Systems Engineering under the supervision of Dr Y. Ahmet Şekercioğlu (main supervisor), Dr. Lachlan Andrew and Professor Tom Drummond (associate supervisors).

**Signed**: _____
Yung Fei Wong (Ricky)

**Date**: August 2017

# Acknowledgments

This PhD journey has been a memorable and invaluable experience for me, and it would not be possible without the assistance and support of many people to whom I am truly grateful.

First, I would like to express deep gratitude to my main supervisor, Dr. Ahmet Şekercioğlu for giving me the opportunity to undertake this PhD research, and most importantly, for offering me important advice on how to be a more independent researcher. Amidst the ups and downs of this journey, his optimism and his ability to instill confidence have been a monumental part in maintaining my enthusiasm in all these years.

I would also like to thank my associate supervisors, Dr. Lachlan Andrew and Professor Tom Drummond. Throughout my candidature, they have played a huge role in making this research possible. In particular, Dr. Andrew has always been there to critique my ideas and made me think more critically about any research problems I have encountered. Moreover, it was always enjoyable to bounce ideas and debate about them with him. For that, I am extremely grateful for the time he spent working with me. Furthermore, I would like to thank him for taking some time off from his busy schedule to proofread a large part of this thesis document and to offer valuable suggestions on improving writing clarity.

On the other hand, Professor Drummond has always been an inspiration to me. His insight and his great mathematical thinking skills have often been the source of motivation to better myself. Especially at the start of my candidature, his vast research experience and his suggestions have been a significant part in providing the needed momentum to push my research in the right direction.

Special thanks should also be given to all staff working at the Department of Electrical and Computer Systems Engineering (ECSE) in Monash University for allowing me to focus on my research while they handle all administrative tasks smoothly. In addition, I am deeply grateful for the support I received from DiUS Computing Pty Ltd during the initial stage of my research, and in particular, I would like to thank Mr. Voon Wong for sharing his engineering expertise on signal processing.

Finally, all of this would not be possible without the emotional support of my friends and family. I would like thank Caesar for listening to my complaints and frustration, and helping me cope with the stresses that are part of this journey, especially in the final year. I am also forever grateful to my parents for having faith in me all this time. Their love and support have always kept me going.

# Contents

# List of Figures

# List of Tables

# Acronyms

| | |
|---|---|
| CELLR | Cumulative Error Log Likelihood Ratio |
| DBN | Dynamic Bayesian Network |
| EDHMM | Explicit Duration Hidden Markov Model |
| EM | Expectation Maximisation |
| HMM | Hidden Markov Model |
| FHMM | Factorial Hidden Markov Model |
| FVTHMM | Factorial Variable Transition Hidden Markov Model |
| MLE | Maximum Likelihood Estimate |
| MML | Minimum Message Length |
| PBDT | Particle-based Distribution Truncation |
| PF | Particle Filter |
| RdFVTHMM | Robust diff-FVTHMM |

# List of Variables

| VARIABLE NAME | DESCRIPTION | UNIT |
|---|---|---|
| $K$ | Number of appliances | – |
| $T$ | Number of time slices | – |
| $x_{t,k}$ | State of the $k$th appliance at time $t$ | – |
| $\mathbf{x}_t$ | $K$-dimensional vector denoting the state of each of the $k$th appliances at time $t$ | – |
| $c_{t,k}$ | Counter variable signifying the current dwelling time of state $x_{t,k}$ at time $t$ | – |
| $\mathbf{c}_t$ | $K$-dimensional vector of $c_{t,k}$, each for the $k$th appliance | – |
| $y_t$ | Aggregate real power measurement at time $t$ | W |
| $z_t$ | Difference signal of the aggregate real power measurement at time $t$ | W |
| $\boldsymbol{\lambda}_\mathrm{e}$ | Parameters of the emission model | – |
| $\boldsymbol{\lambda}_\mathrm{d}$ | Parameters of the state duration model | – |
| $A_k$ | State transition matrix of the $k$th appliance | – |
| $\tilde{A}_k$ | Normalised $A_k$ with self-transition probabilities of zero | – |
| $\mathbf{A}$ | Collection of $A_k$ for all $k$ | – |
| $\tilde{\mathbf{A}}$ | Collection of $\tilde{A}_k$ for all $k$ | – |
| $N_{\mathrm{p,max}}$ | Maximum number of particles per time slice that should be kept | – |
| $M_k$ | Number of states associated with the $k$th appliance | – |
| $\mathcal{S}_t$ | Unordered list of scores associated with the particles at time $t$ | – |
| $\mathcal{X}_t$ | Unordered list of states associated with the particles at time $t$ | – |
| $\psi_t$ | Unordered list of parents associated with the particles at time $t$ | – |
| $\mathcal{C}_t$ | Unordered list of duration counters associated with the particles at time $t$ | – |

CHAPTER ONE

# INTRODUCTION

## 1.1 Motivation

Energy has always been a vital part of human civilisation, and in the modern era, it is more important than ever. Power grids around the world provide the means to power everything from small electronics and household appliances, to big industrial systems that are crucial to the foundation of today's economy and modern life support systems.

Unfortunately, due to population growth and the increased urbanisation in developing countries, global energy usage is projected to continue its upward trajectory, bringing to light the issue of sustainability and resource scarcity. Although the uptake of renewable energy like solar and wind is on the rise, fossil fuel remains the dominant source of electricity globally. According to the U.S. Energy Information Administration's International Outlook 2016 report [U.S16], fossil fuel accounts for 67% of the total world electricity generation in 2012. With consideration of this observation and the growing concerns on climate change, the importance of curtailing energy demand and living sustainably cannot be overstated.

However, it remains a challenge to encourage energy-saving behaviour amongst individuals. Among the many reasons which include habits, an important one is the lack of data on how energy is being used by each appliance. Without the depth of information available, actions that could be taken are limited, if not ineffective. In fact, it is difficult to ascertain the best course of action to take if one were to reduce his/her energy cost based solely on the information on electricity bills; their month-long reporting intervals and the presented *total/aggregate* energy consumption certainly do not help. Even though the recent widespread deployment of smart meters offers improvement by enabling energy information to be accessed in a more timely manner, the energy data pro-

vided is still in the aggregate representation. This forces household occupants to rely on educated guesses and trial-and-errors in identifying sources of high demand. Hence, it is clear that more detailed information in the form of individual appliance energy consumption is needed before specific actions could be taken to effectuate beneficial changes required for sustainable living.

The information can be displayed as a pie chart, showing the proportion of energy used by each appliance. Or better yet, a time progression on how each appliance contributes to the total, and suggestions on the best approach to reduce energy cost without severely impacting comfort levels. One way or another, there are many other benefits afforded by the availability of such appliance-level data, with potential outcomes for stimulating energy saving behaviour among household occupants.

In this thesis, we look at the problem of inferring the energy usage information of appliances, by decomposing the aggregate electrical measurements obtained from a *single* sensing point in a residential setting. The main task involves the development of a new software algorithm for *real-time* disaggregation of the total power consumption data into its appliance-level measurements, thereby allowing the relative proportion of energy used by appliances to be estimated inexpensively and non-intrusively.

## 1.2 Non-Intrusive Load Monitoring

The direct sensing of appliance energy consumption via dedicated sensors is the simplest form of appliance monitoring. Off-the-shelf and ready-to-use hardware can be easily purchased, and setting up is merely a matter of attaching each measurement device to an appliance before connecting them together to form a wireless sensor network (WSN) or a power line communication (PLC) network. Collection of data is then achieved through the transmission of sensed measurements to a central node or a gateway device for logging and analysis purposes.

While the set-up is straightforward and end-use measurements could be accurately obtained, it is not scalable from an installation perspective, given the myriad of sensors that need to be deployed. In and of itself, this may be a sufficient deterrent for majority of household occupants, limiting installations to only those who are already conscious about energy usage. Ideally, it will be more effective to have this appliance-monitoring "service" rolled out by utilities or third-parties, without any effort required on users' part to be involved in the maintenance of the infrastructure. But, clearly, that is not going to work if direct sensing is to be

used, given the aforementioned set-up, cost and labour that would be entailed. Therefore, a less invasive approach to appliance monitoring is required.

Traditionally, less invasive means of obtaining end-use information is performed through an energy audit procedure, where indirect external variables that might affect energy use are employed to infer end-use consumption. These variables may include the thermal insulation capacity of a residential unit, weather data, and information from questionnaires pertaining to household occupants' behaviour, among others [SU09]. However, because only variables external to the electrical measurements of appliances are used, estimates may be less accurate and prone to being biased. This is in addition to the requirement of hiring an energy-auditing expert, which again limits the potential users to those who are already conscious about energy use.

Fortunately, there exists another class of methods that is growing in popularity. Introduced by Hart in the 1980s [Har85], the approach, Non-intrusive Load Monitoring (NILM)[1], assigns detectable patterns in the aggregate-level electrical signals to appliances. These patterns, also known as appliance signatures, allows contributions of appliances to the total to be inferred, as it is assumed that there is a direct correspondence between a particular pattern and a given appliance, much like the one-to-one relationship between a person's handwritten signature and his/her identity. Typically, this association is learned during a training stage, where behaviours of appliances are encoded into usable mathematical descriptions called appliance models.

Considering that only aggregate-level measurements are required in NILM, no dedicated sensor for each appliance needs to be deployed in users' premises. The single-point sensing approach can take on the form of a device attached to a residential unit's circuit-breaker panel or it can be an already-installed smart meter (see Figure 1.1), where electrical values are measured. The measurements are then disaggregated into appliance-level components via a carefully-designed algorithm running on a computation device such as a desktop computer, a server located in the Internet or potentially, even a computationally well-equipped In-Home Display (IHD) unit.

Being cost-effective, non-intrusive and maintenance-free, NILM is thus an appealing approach to tackling the lack of appliance-level energy data needed for stimulating energy-saving behaviour. As such, it will be the subject of study for

[1]NILM is also often synonymously referred to as "load disaggregation" or "Non-intrusive Appliance Load Monitoring (NIALM)" or "single-point sensing".

this thesis, whereby new algorithms are proposed to improve accuracy in the real-time disaggregation of total power consumption measurements.

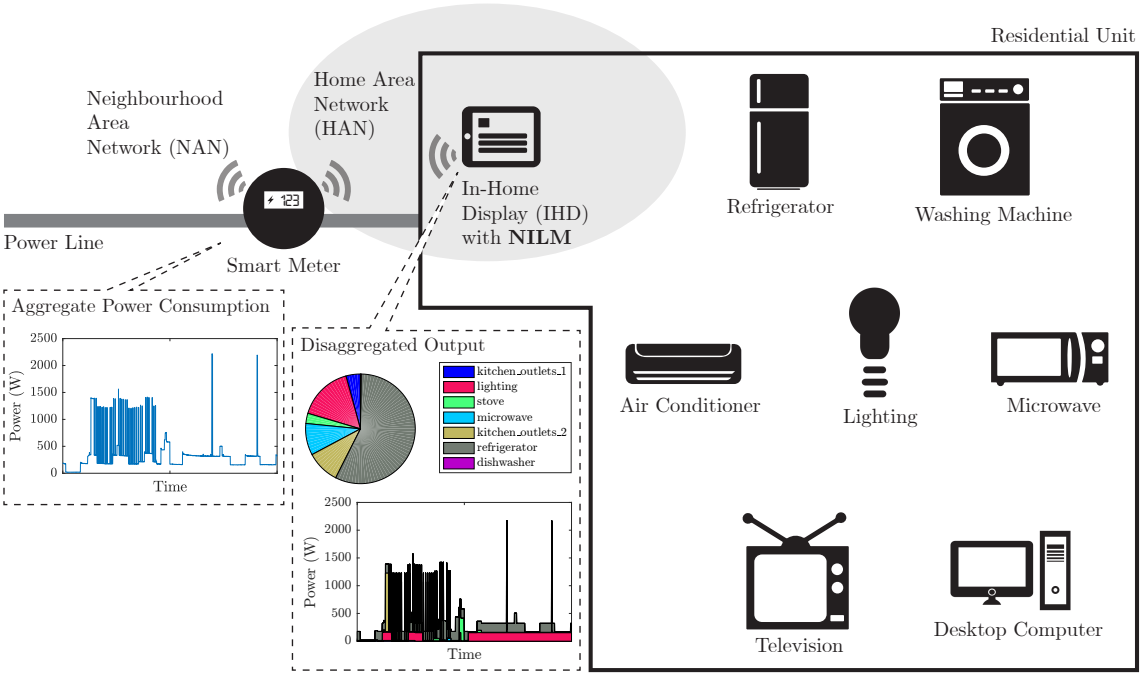



Figure 1.1: A residential unit with NILM.

## 1.3 Potential Applications

The development of NILM has far-reaching applications beyond that of home appliance energy monitoring. Among the prominent ones are:

- **Home Automation Systems**. In home automation systems, the extracted appliance-level data from NILM can serve as inputs for further energy efficiency refinement. For example, with a preconfigured utility bill budget for a given period of time, the latest appliance-level energy information could be used together with the current household occupancy level for dynamic balancing of comfort levels and energy cost. Manual non-optimal controls of appliances are not required as the system adapts to users' behaviours and learns the optimum way to conserve energy. In the worst case where fully automated control of appliances cannot be achieved (due to legacy appliances), targeted suggestions to reach optimality can be given to guide a human operator.

- **Low-cost Occupancy Detection**. Leveraging the single-sensing approach of NILM, low-cost occupancy detection of household occupants can also

realised [CBS+13, TWLX15]. The basic idea is, by monitoring only the aggregate measurements, operational states of common appliances can be inferred and thus, convey current occupancy levels of a residential unit. Such an application might be used as part of smart homes and home automation systems to automatically turn off equipments when no one is at home. However, it is easy to see why this aspect of NILM raises privacy concerns among privacy-conscious individuals, given that the actions of users are closely tied to the operation of appliances.

- **Load-Shed Verification**. One of the main functionalities in smart grids is Demand-Side Management (DSM), whereby control of certain time-shiftable or non-time-critical appliances like dishwashers are voluntarily delegated by household occupants to grid operators in exchange for incentives such as lower billing rates. With the aim of balancing energy supply and demand in the grid, these appliances will be deferred to run at off-peak periods when energy demand is low. However, DSM is built on top of a trust model between grid operators and end-users. If an appliance is tampered to ignore Demand Response (DR) instructions and spoof its real operational state, household occupants can get a free ride on incentives. To counteract this, NILM has been proposed as an inexpensive means to verifying that the shedding of the targeted appliances is successful [BJJ+11].

- **Inexpensive Validation of Energy-related Policies**. Appliance-level data obtained through NILM may also provide a low cost approach to objectively validating the effectiveness of any implemented energy-related policies. For instance, restrictions on the usage of certain appliances could be verified while the estimated appliance-level data pre-policy and post-policy could be compared to gauge the impact of the implemented policy. The transparency afforded by data-driven policies is not only advantageous to policy makers but also to those the policy is imposed on.

- **Equipment Fault Diagnosis**. Naturally, electrical measurements like voltage or current waveforms reveal a lot about the state of devices. For complex devices, access to subcomponents may not be possible. In this regard, being able to non-intrusively probe for deviations from nominal system behaviour allows cost effective diagnosis of faults [DCL+05, SLNC08], which would otherwise not be possible.

## 1.4 Challenges and Research Problems

Despite the promising outlook in numerous applications, NILM is still by itself a challenging problem. One of the main challenges relates to the previously mentioned assumption on the existence of a unique association between an appliance and a detectable pattern induced by the former. Depending on the type of patterns used, it may be common for two or more appliances in a residential unit to have the same appliance signature. Thus, it will be difficult to make a distinction between them.

To get a sense of this, consider a simple case of two appliances, each having a power draw of 100 Watts. Turning on either of these appliances would generate a positive step change of 100 Watts observable in the aggregate power signal. If the change in power is the only type of appliance signature used, it is clearly non-trivial to ascertain which of the two appliances is the true generator of the observed event. Such ambiguity is often the primary cause of poor disaggregation accuracy in the literature.

Another important challenge, rarely addressed by past work, concerns the presence of unknown appliances[2]. These appliances, unseen by a NILM system during the training stage, can be introduced into a residential unit via new purchases or guest visits. Their contributions to the aggregate-level measurements could disrupt the ability of the load disaggregator in performing the intended tasks, with implications of false positives and reduced disaggregation accuracy. For example, patterns induced by an unknown appliance could be wrongly attributed to the closest-matching known appliance in the database.

Given that the introduction of unknown appliances is not uncommon in a real-world setting, a practical NILM system is expected to be able to deal with such situations. At present however, the design of a robust NILM system that can account for this is still a challenge, as it is difficult to differentiate between patterns due to existing appliances and those of unknown appliances. Further compounding this are scenarios where both the former and the latter are similar.

Also of practical concern is the high computational complexity intrinsic to the disaggregation of aggregate-level measurements consisting of a large number of appliances. Being able to reduce computational complexity in such situations is important for interactive applications that demand real-time estimates of appliance-level data (e.g. home automation systems), not to mention a require-

[2]As unknown appliances are appliances that a NILM system has not seen before, models associating electrical patterns and these appliances are non-existent. Therefore, the term "unmodelled appliances" will also be used interchangeably for the remainder of this thesis.

ment for low-latency feedbacks needed for increasing user engagement towards conserving energy.

## 1.5 Existing Solutions

Various state-of-the-art solutions in the literature have been proposed for addressing some of the aforementioned challenges. Here, we only present the summary of trends pertaining to key solutions. For a more complete and expanded discussion, see Chapter 2.

#### Improving disaggregation accuracy

In dealing with similar patterns between different appliances, a common theme amongst recent methods is the use of additional features to aid differentiation. These usually come in the form of high-dimensional electrical patterns (e.g. spectrum of a signal), auxiliary non-electrical data (e.g. from localised motion detectors) or a combination of different features. The main idea is, by using a multitude of features, similarities in one aspect of a pattern can be resolved by differences in another. While the idea is overall beneficial in improving disaggregation accuracy, auxiliary sensors or custom hardware capable of high sampling rates need to be retrofitted, contrary to the minimalistic set-up with only a utility-installed smart meter.

Also separately employed in the state-of-the-art for improving disaggregation accuracy are more complex models for representing appliance behaviours. This means, assumptions are relaxed, allowing learned appliance models to reflect reality more closely. However, this is at the expense of increasing computational complexity, with potential violation of real-time requirements for applications that demand both interactivity and accurate disaggregation. As a case in point, it appears to be a trend amongst existing solutions utilising complex models to employ inherently batch-processed methods (e.g. simulated-annealing) [KJ12, KAL11]. Ideally, it would be of interest to resolve the conflicting need for better models and real-time computation via carefully-designed heuristics and better structural representations of candidate estimates.

#### Dealing with unknown appliances

Unfortunately, there are limited investigations done explicitly for extracting contributions of known appliances in the presence of unknown loads. A thorough

search in the literature either reveals little mention of accounting for such situations, or unrealistically assumes that every appliance in a residential unit can be modelled. For the two that do address the problem [KJ12, TWLT16], an additional noise component representing unknown appliances, together with prior assumptions, are included in the model for disaggregation. However, both approaches are inherently batch-processed, with limited provisions for real-time applications.

## 1.6 Research Objectives

With consideration of the general limitations outlined in the previous section, it is clear that there remains an open question as to whether a robust real-time disaggregation approach, without the luxury of custom hardware, is feasible. To that end, we will constrain our attention to the following aims:

- Resolution of similarities in electrical features induced by different appliances without the need for high frequency (kHz) data or auxiliary non-electrical measurements.
- Real-time disaggregation of electrical measurements, representative of those obtainable from the home area network (HAN) interface of smart meters.
- Robust extraction of power contributions of known appliances even in the presence of unknown loads.

Each of these forms the core component of this research, where solutions are proposed for related problems in stages, before being integrated into a framework usable for disaggregating aggregate-level data of real houses.

## 1.7 Contributions

In meeting the stipulated objectives, the research has resulted in the following main contributions:

- **Appliance Model with Time-varying State Transition Information**. Underlying each appliance in a broader class of latent variable models is an internal state (e.g. ON, OFF etc.) that determines the appliance's electrical measurements. While this class of representations is natural and its use underlies some of the best-performing NILM approaches in recent years,

assumptions on state dependencies have to be made for tractable computations. Common realisations are based on hidden Markov models (HMM) [PGWR12, KJ12, EBE15, MPB+16] and to a lesser extent, instances from a general class of hidden semi-Markov models (HSMM) [KAL11]. In our work, we proposed a new variant of the latter for NILM, adapted from the field of acoustic speech modelling [Vas91, RW92], for enabling the separation of appliances with similar signatures, and the incremental calculation of time-varying duration-dependent state transition information.

- **Method for Fast Inference of Appliances' States**. Like any appliance models based on factorial HMM (FHMM) [GJ97], inferring the unknown states of appliances under the proposed model, given the aggregate-level measurements, is inherently a computationally-intensive task; the space of solutions grows exponentially with the number of considered appliances, with additional complexities pertaining to the inclusion of state durations in the model. To that end, we proposed a novel fast real-time approximation algorithm with good convergence properties, while borrowing the dynamic programming approach of the Viterbi algorithm and the survival-of-the-fittest principle from particle filters. The algorithm generates plausible candidate solutions called "particles" as new aggregate measurements arrive, before truncating away those which are highly unlikely given current circumstances, historical measurements and past estimates. In the implementation, the distribution of particles at each discrete time step is exploited to accelerate computation, enabling a further speed improvement of up to 20 times over a direct implementation. Altogether, this contributes to an empirically linear growth of time complexity as the number of appliances increases. The proposed method is thus a computationally scalable approach for real-time disaggregation.

- **Appliance Model for Slowly-decaying Features**. The power consumption for some appliances (e.g. refrigerators) gradually decreases from the peak at the onset of the turn-on period to a nominal value, as the appliance remains in the same state. To account for these slowly-decaying features, the proposed appliance model with time-varying state transition information is extended to further improve disaggregation accuracy and the model's representation of reality. The result is a specific instance of a segmental HMM used in speech pattern modelling [Rus93], with inherited benefits of the initially proposed model. In terms of model representation, the power con-

sumption for a given state is no longer assumed to be stationary; its mean is modelled to vary according to the state dwell time. This formulation extends the flexibility of the initial model without requiring any baseline changes, thus largely maintaining compatibility with the proposed state inference method mentioned previously.

- **Integrated Real-time Disaggregation Framework with Robust Noise Model**. Inspired by the work of Kolter and Jaakkola [KJ12] alluded to in Section 1.5, the first two contributions are augmented with a noise model based on compressed sensing to form a robust real-time disaggregation framework. It will be used for extracting power consumption of known appliances without being severely affected by the existence of unknown loads. To also guard against the problem related to spurious spikes in power, a steady-state segmentation algorithm based on Hart's work [Har85] is used. Altogether, the framework allows for models of unknown appliances to be left unspecified, forming an alternative means to keeping computational complexity low, while enhancing the robustness of the real-time disaggregation process against the effects of unknown loads.

## 1.8 Organisation of the Thesis

The remainder of this thesis is structured as follows.

**Chapter 2: Non-intrusive Load Monitoring: A Review**

Chapter 2 presents a review of the methods for Non-intrusive Load Monitoring (NILM). The different stages of a typical NILM system are first outlined. Then, details relevant to each stage are described, with particular emphasis on the types of appliance signatures, appliance models and load disaggregation techniques as used in the literature. With the outcome of the review, a baseline objective and a more detailed research scope of this thesis are presented.

**Chapter 3: Modelling of Appliance Behaviour**

Chapter 3 begins with a general overview on related approaches taken for modelling appliance behaviour. Then, with consideration of the goal of this research, a new model with dynamic state transition probabilities based on state dwell time is proposed for NILM. This is followed by details pertaining to the learning of parameters for the new model, where a robust procedure and a method based on an

information criterion – minimum message length (MML) – are presented. Subsequently, the proposed model is justified and validated through an experiment to show how different appliances which consume the same power consumption can be identified correctly from the aggregate measurements, while also demonstrating the value of the dynamic state transition probabilities in aiding incremental calculations of probability values needed for the real-time state inference method described in subsequent chapters.

**Chapter 4: Appliance State Inference**

The problem of appliance state inference under the proposed appliance model is explored in Chapter 4. Potential solutions such as the Viterbi algorithm and alternative approximation methods are described from a computational complexity standpoint and in terms of their ability to meet real-time requirements. Then, in light of their limitations, the chapter presents a new computationally efficient state inference algorithm – Particle-based Distribution Truncation (PBDT). In the end, experimental results of applying the devised method on a public dataset of real homes are shown and discussed.

**Chapter 5: Robust Extraction of Appliance Power**

In Chapter 5, we consider the more challenging task of load disaggregation in the presence of unmodelled appliances, unlike in previous chapters. Specifically, special attention is devoted to the development of techniques which are resilient against the effects of unknown devices in homes, enabling the power contributions of modelled appliances to be extracted robustly. Implications of not taking into account unknown loads are demonstrated, and extensions to the techniques as discussed in Chapter 3 and Chapter 4 are detailed. In closing, using the power measurements of real homes as test data, the chapter provides a quantitative analysis on the method's extraction ability and its robustness.

**Chapter 6: Conclusion and Future Work**

Finally, Chapter 6 summarizes the key contributions of the thesis and discusses potential future directions for this research.

## 1.9 Publications

During the course of this research, the following peer-reviewed publications have been made:

- Y. Wong, Y. A. Şekercioğlu, T. Drummond, V. Wong, "Recent Approaches to Non-Intrusive Load Monitoring Techniques in Residential Settings", IEEE Symposium on Computational Intelligence Applications In Smart Grid (CIASG), pp. 73-79, 2013. [WcDW13]
- V. Wong, Y. Wong, Y. A. Şekercioğlu. T. Drummond, "A Fast Multiple Appliance Detection Algorithm for Non-Intrusive Load Monitoring", IEEE Symposium on Computational Intelligence Applications In Smart Grid (CIASG), pp. 80-86, 2013. [WWDc13]
- Y. Wong, Y. A. Şekercioğlu, T. Drummond, "Real-Time Load Disaggregation Algorithm using Particle-based Distribution Truncation with State Occupancy Model", IET Electronics Letters, pp. 697-699, May 2014. [WcD14]

The following additional publications based on new content in Chapter 4 and Chapter 5 are under preparation:

- A Fast State Inference Algorithm for Real-Time Load Disaggregation (To be submitted to: ACM Journal of Experimental Algorithmics)
- A Robust Method for the Real-Time Disaggregation of Smart Meter Data (To be submitted to: IEEE Transactions on Smart Grid)

CHAPTER TWO

# NON-INTRUSIVE LOAD MONITORING: A REVIEW

This chapter provides a background on the state-of-the-art approaches in Non-intrusive Load Monitoring (NILM). We first present an overview of the key design considerations of a typical NILM system, with particular attention devoted to the behaviour of different types of appliances, electrical signal acquisition, common features used for discriminating between appliances, modelling techniques and methods for performing disaggregation. The discussion centres on the main issues presented in Section 1.4 and expands on the brief treatment of existing solutions given in Section 1.5. Then, in relation to the robustness requirement and the real-time requirement in disaggregating smart meter data, we present a summary of limitations in the literature. This provides the motivation of the work presented in this thesis. Also, to facilitate the discourse on the proposed methods in the subsequent chapters, a review on the theory of Bayesian graphical models, hidden Markov models and standard approaches for performing state inference is given. Parts of the chapter related to the review of NILM have been published to a conference paper [WcDW13]. For a different outlook, the reader can refer to two other comprehensive surveys on load disaggregation – [ZR11a] and [ZGIR12]. In closing, a summary of limitations in existing approaches is given, before descriptions pertaining to public energy consumption datasets of real homes which can be used for benchmarking are provided.

## 2.1 Introduction

As described in Chapter 1, Non-intrusive Load Monitoring (NILM) or load disaggregation is a class of techniques used for estimating the component signals attributable to individual appliances, given only the availability of a composite

signal. In this way, NILM is similar to a blind source separation (BSS) problem [Car98]. Although it has been nearly three decades since its introduction by Hart [Har85], research in this area has been slow until recent years. The present uptake of NILM in the research community is largely driven by the potential integration with smart grid systems, the improved computational throughput enabled by new hardware technologies and the recent publications of datasets pertaining to household and appliance measurements, among others.

Generally, the process of NILM consists of four main stages. As shown in Figure 2.1, they are: (1) Electrical Signal Acquisition, (2) Feature Extraction, (3) Appliance Modelling and (4) Appliance Identification. The first stage involves the sampling of aggregate electrical signals such as voltage and current waveforms using Analog-to-Digital converters (ADC). Then, electrical features are extracted from the sampled signals, before their association with the operational state of appliances are learned and encoded into appliance models as part of a training procedure. Finally, at disaggregation time, the extracted features are classified according to the learned models to identify the contributing appliance.

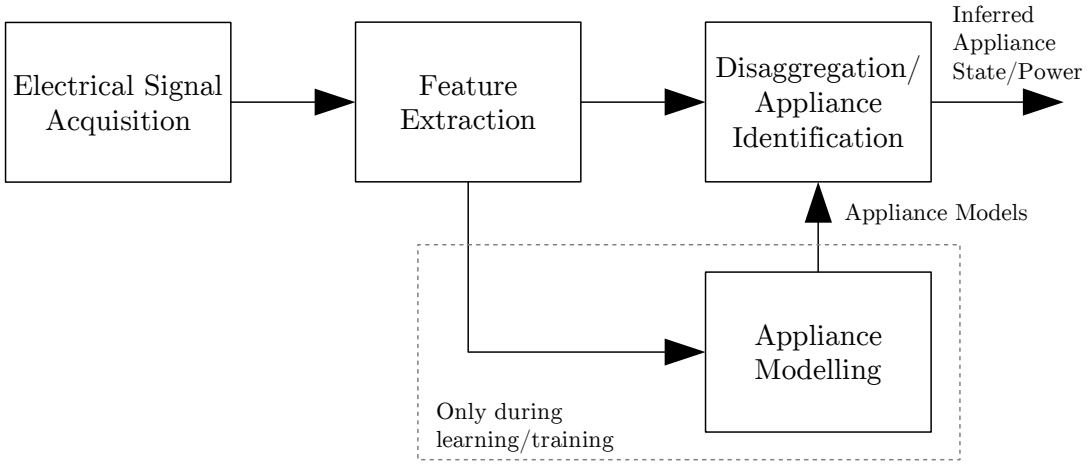



Figure 2.1: Stages of NILM.

It turns out that techniques developed in the past are mainly variations in terms of these four stages, with each stage intricately linked to the others. For example, the type of acquisition hardware determines the features that can be extracted and the employed features govern the appropriate types of models that could be used. This in turn controls the nature of the NILM algorithm and the associated appliance recognition scheme. As such, subsequent discussions on existing approaches will be made in terms of these aspects. However, before describing the core aspects of NILM, we will first provide a brief treatment of the

different kinds of appliances in the next section to put into perspective the operational differences that could be exploited for aiding disaggregation.

## 2.2 Types of Appliances

Appliances can take on many forms. Some can be working independently in the background without any user interaction, while others are directly controlled by users. In this regard, the detection of appliances in the aggregate signal warrants a thorough understanding on the operational behaviours of the different classes of appliances in the real-world setting. Four categories of loads can in general be defined according to Hart [Har92] and Zeifman [ZR11a]:

- **Class-I**: Binary-state appliances that operate in two distinct ON/OFF states such as lights and toasters;
- **Class-II**: Multi-state appliances with finite state machines (FSM) representation like refrigerators and washing machines. For example, the former could transition from the ON state where the compressor is running to the state corresponding to the defrost cycle;
- **Class-III**: Continuously variable loads which draw power in a continuous non-discrete manner such as light dimmers and variable-speed drives (VSD) appliances (e.g. power drills and commercial Heating Ventilation and Air Conditioning (HVAC) systems). This group is more common in industrial and commercial settings [RLBH94];
- **Class-IV**: Always-on appliances with constant power consumption like household security cameras and wireless routers. Appliances of this kind are not manually toggled by users and they run in the background 24/7.

In terms of the challenges in disaggregation, Class-I and Class-II are relatively easy to detect. But, because they make up the bulk of appliances in a residential unit, disaggregation errors might occur due to the more likely non-unique associations between electrical patterns and appliances. Even though Class-IV devices may not contribute much to the overall energy consumption, they are typically unmodelled and notorious for shifting the aggregate power signal upwards, thus severely affecting the extraction of modelled loads from the aggregate measurements. To date, Class-III appliances remain the most challenging ones to extract, given their on-demand continuous variation and the potentially non-repeating behaviour.

An alternative characterisation of the different classes of appliances has also been given by Sultanem [Sul91]. However, they are grouped by the electrical components embedded in devices as follows:

- **Resistive appliances**: Examples in this class are dominantly resistive with little or no capacitive or inductive elements. Heaters and incandescent lighting belong to this category. Due to the resistive nature, only real power[1] is drawn from the mains in theory and transient electrical patterns at the onset of energisation are virtually non-existent;

- **Pump-operated appliances**: Appliances of this kind have pumps driven by electric motors. Common examples include refrigerators, dishwashers and washing-machine drain pumps. The inductive nature of the electric motors means reactive power will be drawn. This is on top of the inducement of odd-numbered current harmonics during operation, and the prominent and lengthy transients when these appliances are switched on;

- **Motor-driven appliances**: The operational behaviour of this class of loads is similar to the pump-operated ones, but differ in that the transients are less salient at the onset of energisation. Examples are appliances with motors only, such as fans and electric mixers;

- **Electronically-fed appliances**: These appliances may refer to those which are powered via switched-mode power supplies (SMPS). Due to the use of high-frequency switching for output voltage regulation, loads belonging to this class induce high-frequency noise on the electrical line. Hence, they are generally associated with the rich spectral content in the kHz-MHz range when being operated and the short highly salient transients at the onset of turning on [PRK+07, GRP10]. Prominent examples of this class are personal computers, televisions and mobile phone chargers;

- **Electronic power control appliances**: Despite being mentioned, Sultanem did not provide a clear description on this category. However, judging from his remark on the dependence of load characteristics on the operating power level, this class might refer to the group of appliances which are internally controlled using proportional-integral-derivative (PID) controllers

[1] It is colloquially called ”power” and measured in Watts, but in electrical engineering, the technical term is ”real power” or ”active power”. This is to distinguish it from another related quantity named ”reactive power” which is measured in the unit Vars. In this thesis, whenever ”power” is used without the quantifiers ”real” or ”reactive”, the former is implied. The formal definitions of both real power and reactive power are given in Section 2.3.2.

with a feedback mechanism. In this way, it may include the previously mentioned Class-III appliances;

- **Fluorescent lighting**: Lighting of this variant is in its own category according to Sultanem [Sul91]. This is possibly due to the intrinsic two-stage transients when powered on. Also characteristic of this class are the high third-order harmonic content embedded in the current waveform and the large phase difference between the voltage and current signals. Though, modern fluorescent lamps with electronic ballasts could additionally induce high-frequency noise on the electrical line, owing to the use of switching electronics like those for SMPS [GRP10].

Regardless of the rules used for grouping, there exists different effective strategies for correctly detecting the disparate classes of appliances from the aggregate measurements. For example, an appliance which induce large harmonic content onto the electrical line may be best identified using spectrum features and appliances with distinctive transient patterns during start-up could be better off detected using shape attributes. A more detailed description on the type of electrical signal acquisition and the various extractable features used in existing approaches are presented in the next and subsequent sections.

## 2.3 Electrical Signal Acquisition

The first stage of NILM is the acquisition of electrical data. Depending on the appliance detection requirement at hand and the nature of the algorithm used for disaggregation, the data could be fundamental electrical measurements such as voltage and current waveforms, or derived electrical quantities like real power, root mean square voltage ($V_{\text{rms}}$) and root mean square current ($I_{\text{rms}}$). One way or another, the data can be dichotomised into two types: high sampling rate and low sampling rate. In this section, we present the details of each type and illustrate with examples from past works.

### 2.3.1 High Sampling Rate

Voltage and current waveforms are inherently high sampling rate data which are typically represented by digital samples obtained from Analog-to-Digital converters (ADC). To meet the signal reconstruction requirement stipulated by the

Nyquist-Shannon sampling theorem [Jer77], the sampling rate in the digital conversion process should be at least twice the highest frequency content of the analogue signal. Given that the voltage signal at the power outlet of a residential unit (in Australia) is a 50Hz sinusoidal waveform with $V_{\text{rms}}$ of 240V, voltage samples have to be collected at a minimum rate of 100Hz, whereas the current signal is load-dependent; if the load is linear, the waveform is sinusoidal with the same frequency. But, when non-linear loads are present (e.g. SMPS), non-sinusoidal current with high order harmonic frequency content will be drawn. Altogether, this means sampling rate in the kHz range or more is required for capturing salient features. For example, Patel et al. [PRK+07] utilised a data acquisition module with a sampling rate of 100MHz to capture high frequency noise introduced on the electrical line by appliances, while Kolter and Johnson [KJ11], in their collection of raw voltage and current samples for disaggregation, employed sampling rates of 15kHz.

In practice however, the direct usage of raw data is not common in NILM. Rather, spectrum-based quantities from high sampling rate data are more frequently used. These can come in the form of Fourier Transforms (FT) or Wavelet Transforms (WT). For FT, the frequency representations of both voltage and current are

$$V_{\text{f}}[k] = \sum_{n=0}^{N-1} v[n] \exp^{-j2\pi kn/N} \tag{2.1}$$

$$I_{\text{f}}[k] = \sum_{n=0}^{N-1} i[n] \exp^{-j2\pi kn/N}, \tag{2.2}$$

where $v[n]$ and $i[n]$ denote the $n$th sample value of the voltage and current waveform respectively, $j$ is the imaginary unit used in complex numbers, $k$ is the index of the frequency component, and $N$ refers to the number of consecutive samples used for the computation. The outcomes are high-dimensional feature vectors, $V_{\text{f}}$ and $I_{\text{f}}$, computed at every $N$ samples, which could be used as the basis for appliance identification.

While FT is just one example, the availability of high sampling rate fundamental electrical data allows numerous other derived electrical quantities to be computed (see Section 2.4 for details). This means little or no restrictions are imposed on the NILM algorithm designer as he/she could possibly employ any appropriate features from the waveform data to identify the different types of loads discussed in Section 2.2. In fact, it has even been shown that a myriad of quantities can be used in tandem as features of higher dimensionality to boost

disaggregation accuracy [LNKC10a]. Though, the downside is the need for non-consumer-friendly custom hardware capable of high sampling rates and an elaborate monitoring set-up for handling the naturally high influx of data. Either way, as we shall see in the next section, low sampling rate data of certain derived electrical quantities can be easily computed from high sampling rate raw voltage and current measurements but not vice versa.

### 2.3.2 Low Sampling Rate

We consider low sampling rate data as those measured at rates below ∼1Hz using standard consumer-focused instrumentation equipments such as smart meters and off-the-shelf metering devices. Measurements like hourly aggregate energy consumption (in kWh) and power consumption (in Watts) reported in sub-Hz rates are typical examples[2]. While high sampling rate electrical data are technically employed by these devices for internal calculation of the aforementioned quantities, they are inaccessible for external use (e.g. feature extraction). Therefore, low sampling rate derived quantities from such equipments have to be used as it is for disaggregation.

Perhaps, the most common form of data utilised in this context by NILM researchers in recent years are real power and to a lesser extent, reactive power. For a given time interval $T$, they are both defined as

$$P(t) = \frac{1}{T}\int_{t-T}^{t} v(\tau)i(\tau)d\tau \tag{2.3}$$

$$Q(t) = \frac{1}{T}\int_{t-T}^{t} v(\tau)i\left(\tau - \frac{1}{4f}\right)d\tau, \tag{2.4}$$

where $P(t)$ and $Q(t)$ are the real power and reactive power for the time interval starting at $t-T$ and ending at $t$; $v(\tau)$ and $i(\tau)$ are the voltage and current signal value at time $\tau$ respectively; and $i\left(\tau - \frac{1}{4f}\right)$ is $i(\tau)$ shifted by $\pi/4$ radians relative to $v(\tau)$, with $f$ being the fundamental frequency of $v(\tau)$ (e.g. $f = 50$ in Australia).

Being derived quantities which are computed at relatively coarse time intervals (i.e. with the value of $T$ corresponding to sub-Hz rates), fine-grained observable electrical properties of high frequencies are lost. As such, robust NILM algorithms utilising only low rate power-related measurements are more challenging to design correctly. Indeed, any proposed solution has to contend with

[2]The Australian Smart Metering Infrastructure Minimum Specification [NSM11] mandates that a smart meter should be capable of reporting real power measurements up to a rate of 0.2Hz.

the well-known issue of similarities in power consumption between appliances; otherwise, ambiguous situations with a certain change-in-power value shareable among multiple appliances are non-trivial to resolve. Also, the often used one-at-a-time assumption [Har92] for reducing computational requirements, where only one appliance is able to change state in one discrete time step, does not generally hold at low sampling rates.

Yet, despite the innate difficulties, the disaggregation of low rate electrical data has been predominantly the research focus in NILM lately. This is particularly because of the many intrinsic benefits, which include the reusing of existing monitoring infrastructure (e.g. smart meters), the cheaper set-up given the generally low volume of data, and the low complexity deployment procedures from the perspective of both end-users and utilities. As such, the widespread deployment of NILM systems using only low sampling rate data is more achievable as compared to those based on high sampling rate data, and in this thesis, we will focus our attention on the former, where only low rate (e.g. $\sim$1Hz) real power measurements are available.

## 2.4 Appliance Signatures

Following the acquisition of aggregate electrical data, of interest are the extractable features that could be used as signatures for appliance identification. While the association between features and appliances should be unique in the ideal case, it is often difficult to achieve in reality, given the large number of appliances in a typical residential unit and the potential operational similarities between different appliances. For these reasons, the choice of appliance signatures remains a key design consideration in NILM, especially for situations where raw high sampling rate voltage and current waveform data are not obtainable.

In this section, we present a survey of the various appliance signatures used in existing approaches with respect to a categorisation shown in Figure 2.2, while also discussing how they relate to the task of disaggregation in the context of the aforementioned uniqueness problem. Appliance signatures belonging to the steady-state category are first introduced, before those in the transient and non-traditional categories are discussed.

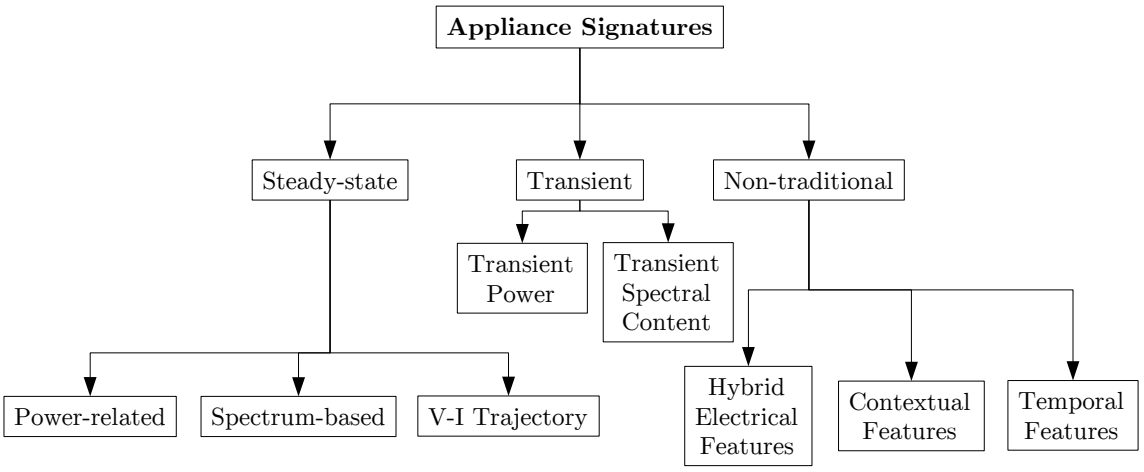



Figure 2.2: Categorisation of appliance signatures.

### 2.4.1 Steady-State Signatures

Steady-state signatures are persistent features which are detectable from the aggregate measurements during the steady-state operation of appliances. As the signatures are naturally stable for long time periods, data with small reporting intervals is generally not required, although high sampling rate measurements of voltage and current waveform are still needed when derived electrical quantities, used as part of steady-state signatures, are not provided by the metering device. Features based on power-related measurements are examples of the former, since, as mentioned in Section 2.3.2, most consumer-focused instrumentation equipments could record real power consumption. To that end, steady-state power-related quantities are one of the most popular types of signatures used in the recent years.

In the seminal work by Hart [Har92], step changes in steady-state aggregate real power, $\Delta P$, and step changes in steady-state aggregate reactive power, $\Delta Q$, are used as the basis for appliance identification. The premise is, if an appliance is switched on at a certain time, it is expected that at least one of $\Delta P$ and $\Delta Q$ will be non-zero. In the case where historical values of $\Delta P$ and $\Delta Q$ corresponding to an appliance have been characterised, future observations of similar values can be inferred to be coming from the same appliance. For example, if a given appliance is known to consume 200W and 100Var during steady-state operation based on some prior knowledge, then any future observed perturbations in the ballpark of +200W and +100Var are likely due to that appliance being switched on. Likewise, step changes of similar magnitudes in the negative direction could be attributed to turn-off events of the same appliance. In this respect, the pair $(\Delta P, \Delta Q)$ can be treated as a two-dimensional signature that could be visualised

in a 2D map shown in Figure 2.3, and each newly observed $(\Delta P, \Delta Q)$ is estimated to be arising from the closest cluster based on some distance metric. Though, in doing so, care must be given to the extraction of the steady-state values of power quantities, considering that appliances, especially those with high inductance or capacitance, are notorious for introducing transients in the power consumption signal (e.g. surge in power) when turned on. Hence, in his implementation, Hart has included an algorithm for disregarding non-stable values when computing $\Delta P$ and $\Delta Q$.

While the idea is overall simple, Hart's prototype, like others based on clustering in the $\Delta P$-$\Delta Q$ space [CA98a, MHHE11], has one main issue. That is, it is difficult to distinguish between appliances with similar values of $(\Delta P, \Delta Q)$. The problem is especially profound for the case of low power loads since many household appliances fall within this range, as can be seen in the bottom left corner of Figure 2.3 where many cluster centroids overlap with one another. In fact, from

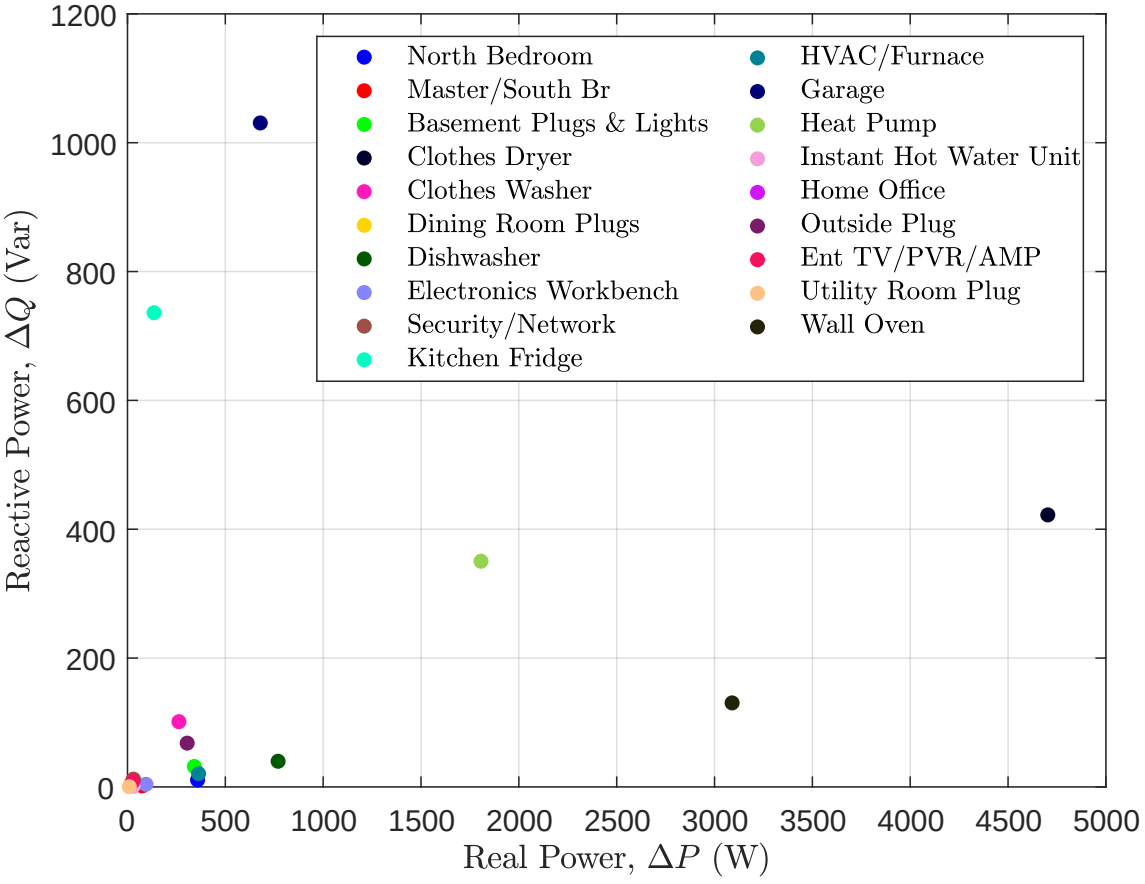



Figure 2.3: Centroids of clusters associated with the ON state of appliances. The appliance data used for generating this figure is from the AMPds dataset [MPB+13]. For the purpose of illustration, each appliance is treated as only having one ON state, even though appliances could have multiple ON states (e.g. different settings) in reality.

Hart's own experiments [Har92], appliances with power consumption below approximately 150W could not be extracted reliably.

To counteract this problem, steady-state spectrum-based signatures have thus been proposed as follow-up improvements. For instance, Cole and Albicki [**?**] explored the use of higher order harmonics associated with the current signal via means of a 256-point Fast Fourier Transform (FFT). He found that the harmonic content induced by loads during steady-state operation is sufficiently repeatable between measurements to warrant its use as an appliance signature. Although not implemented as a complete system, he further suggested the utilisation of both steady-state power level changes and higher order harmonic content for load identification.

Following this is the work of Laughman and his colleagues [LKC+03]. They augmented the existing $\Delta P$-$\Delta Q$ space with additional dimensions representing the higher-order harmonic components of the current signal, and they showed that by including the change in steady-state third order harmonic content as an extra feature, overlaps in the original $\Delta P$-$\Delta Q$ space can be resolved. The use of spectral features computed using FFT in this way has motivated others in later work to do the same, even though some only used spectrum-based signatures without regard for features in the original $\Delta P$-$\Delta Q$ space [SNL06, PRK+07, GRP10, LWP12].

In particular, Patel et al. [PRK+07] and Gupta et al. [GRP10] exploited the continuous Electromagnetic Interference (EMI) voltage noise naturally induced by certain appliances (e.g. SMPS) on the electrical line during steady-state operation. Their set-up consist of a data acquisition hardware capable of capturing spectral content in the frequency range of 36kHz to 500kHz, which allows them to profile the operational state of household appliances using high-dimensional feature vectors obtained from frequency transforms of the time-domain voltage signal. Concerns of overlaps in the new feature space have been validated and it was found that spectral features in the high frequency range is especially distinct among different brands of appliances. Further, for cases with multiple appliances with the same brand, they also discovered that distinction could still be made due to variation in the spectral content as a result of manufacturing variability of household appliances.

Despite achieving detection accuracies of $\sim$90%, there are a number of limitations and open questions. First, it is not clear how appliances which do not continuously emit significant EMI are to be detected. Examples, as remarked by the authors [GRP10], are resistive appliances like dryers and stoves; and old loads

without switch-mode power supplies. Second, as only spectral features are used, appliances could be detected but their energy consumption cannot be estimated. Also noted by Zeifman and Roth [ZR11a] in their own survey is the possible dependence on the topology of the electrical network of a house, given that the observed spectral content could in addition be a function of the stray inductance or capacitance of wirings. This means signatures may differ depending on where in the electrical network the EMI is measured.

Apart from the aforementioned features, there also exists one other unconventional type of steady-state signature based on raw voltage and current waveforms. First investigated by Lam et al. [LFL07], the concept depends on the use of the phase trajectory of the voltage signal (V) with respect to that of the current signal (I). The trajectory, when plotted and visualised as a locus in the V-I plane over one complete waveform cycle (see Figure 2.4), can be treated as a feature vector. It was found that appliances of the same type have similar V-I trajectories, allowing appliances to be grouped together objectively according to the component-level category discussed in Section 2.2. While the feature is appealing and may resolve overlaps in $\Delta P$-$\Delta Q$ space, the authors only frame it in the context of building a taxonomy of electrical appliances; load disaggregation using extracted V-I trajectory was not tested.

In summary, from the standpoint of distinguishing between similar appliances, it is apparent that steady-state signatures in the form of spectrum-based features and raw waveform trajectory could outperform power-related features

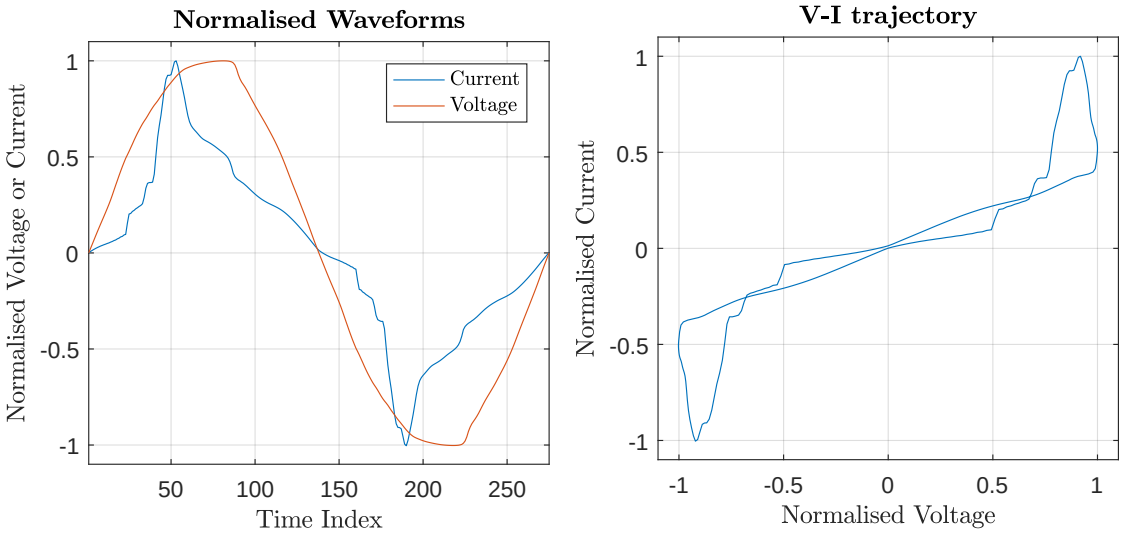



Figure 2.4: Normalised voltage and current waveform over one cycle and the corresponding V-I trajectory. The data used for generating this figure is from house 3 of the REDD dataset [KJ11].

alone. However, due to the need for raw waveform data when deriving high-dimensional features, and the requirement of non-consumer-focused instrumentation devices capable of high sampling rates, they do not conform to our research objectives as stipulated in Section 1.6. Therefore, power-related features remain the focus of our research, but attention is given to new intelligent ways to resolve overlaps without the use of raw waveform data. Nevertheless, one important point should be made with regards of the class of steady-state signatures. That is, they are generally incapable of tracking Class-III loads (e.g. variable loads) as appliances belonging to this category do not commonly change power consumption in discrete levels.

### 2.4.2 Transient Signatures

Closely tied to the nature or type of appliances, transients are non-stable electrical patterns which can be observed when appliances switch from one state to another (e.g. OFF to ON). Common examples are higher-than-rated consumption of power at times when motor-based appliances are first turned on, followed by gradual decrease to steady-state values; this is an effect that is attributed to the increase in electrical impedance as the motor spins up to its nominal rotational speed. Hence, despite acting as nuisances in the extraction of steady-state features, transients are especially suitable to be appliance signatures for they characterise the behaviour of appliances at the electrical level.

Being the foundation for most transient-based techniques that follow [NL96, KLL97, LKC+03], one of the first prominent works in this area was done by Leeb et al. [LKLS93, LSK95]. In their approach, a multi-scale prototype transient event detector for NILM was developed to map transients and the associated time progression of spectral content of the current waveform to the corresponding appliances. The outcome is a tree-structured decomposition scheme to reduce the amount of training required for each appliance, given that only one appliance signature is needed to summarise the operational characteristics of a class of related appliances. For instance, the general transient shapes of various induction motors are similar; the magnitude and duration are just a scaled version of one another. Using this observation, it was shown that applying a stored transient signature of a motor to identify a motor of different brand which is not recorded in the database is possible.

As an extension to the aforementioned work, Cox et al. [CLSN06] built upon the existing detector to investigate transients related to line voltage distortions

when appliances change states. In contrast to the previous method, no current measurements is needed; instead, the spectral envelope of both live-to-neutral and neutral-to-ground voltages are computed before any extracted transient features are classified. This means any power outlet in the house can be conveniently used as a sensing point to detect the operational states of all appliances. Interestingly, the use of spectral content from voltage transients in this regard mirrors the later work by Patel et al. [PRK+07] except that Patel and colleagues utilised wideband bursty EMI signals in the high frequency range at times when appliances are switched on, in addition to the steady-state continuous EMI described in Section 2.4.1. Chang et al. [CYL08] later followed up with an alternative approach by introducing their own turn-on transient energy detector. The detector is based on an iterative algorithm that computes the energy associated with a transient event, thus providing a different way of identifying appliances of interest.

Overall, transient signatures are richer in information compared to their steady-state counterparts. However, there are some issues that impede their widespread use. Firstly, switching events that are in close proximity with one another in time would produce composite/overlapping transient trajectories which are difficult to be separated [NL96]. Secondly, although turn-on events are often investigated for transients, mapping turn-off events is hard, given that on-to-off transitions do not typically generate transient patterns in most cases [Har92]. And lastly, there is the inevitable requirement of using high sampling rates to capture significant transient patterns. Unless a separate, more capable add-on device is installed at the metering point, the use of transients as sole discriminants in load disaggregation is limited when only smart meter data is available.

### 2.4.3 Hybrid and Non-Traditional Signatures

Apart from the two main groups of appliance signatures, there exists some others which do not fit into the categorisation of steady-state and transient. These signatures are often not used in a standalone manner as they serve to augment those which have been discussed in the previous two subsections. Broadly, they can be subdivided into those that are combinations of steady-state and transient electrical patterns, those that are augmented with temporal features and those that exploit contextual features such as temperature.

### Hybrid Electrical Features

The combined use of both steady-state and transient signatures is a natural development, considering that some electrical features are more prominent for one class of appliances but not the others. Case in point, from Section 2.2 and the discussion provided by Sultanem [Sul91], highly resistive loads like heaters do not commonly show distinctive transient characteristics while other types of appliances with inductive or capacitive elements do. Therefore, it is difficult to separate a set of resistive appliances with transient features alone. However, by using transients as discriminants in the first step of a disaggregation process, resistive appliances could be distinguished from those which are non-resistive before separation within the resistive class is made based on other signature types such as $\Delta P$ and $\Delta Q$. Similar arguments in favour of this combined approach have been suggested by Norford et al. [NL96] and Laughman et al. [LKC+03].

The work of Laughman et al. [LKC+03] has been noted in Section 2.4.1 for improving upon the use of steady-state power-level changes by means of steady-state spectrum-based features. On top of that, they also included the analysis of transient signatures adapted from [LSK95] as part of their exploration. While only plans were mentioned for further integration work, their preliminary investigation revealed that the combination of spectrum-based features and transient signatures allows Class-III loads (e.g. variable-speed drives (VSD) loads) to be tracked non-intrusively. Specifically, by exploiting the similarities in variation between the time progression of real power and the harmonic content of the current signal, and by assuming that the harmonic content of a certain order is largely generated by VSD loads but no other appliances, the variable component of the aggregate power consumption can be extracted and removed. This allows the leftover signal to be disaggregated using standard techniques based on steady-state signatures.

Also relevant to the use of hybrid electrical features is the approach proposed by Liang et al. [LNKC10a, LNKC10b]. Their work is unique in that it is seemingly the first to include more than a few types of electrical patterns in a formal disaggregation framework. In particular, raw current waveforms, real and reactive power, harmonic content in current signals, instantaneous admittance waveform, instantaneous power signal, eigenvalues of current waveforms and switching transient waveform are all extracted to jointly contribute to the disaggregation process. From their experiments, it was revealed that the combined approach outperformed all cases where only a single feature type is utilised. Unfortunately, due to the dependence on raw high sampling rate voltage and current

waveform data, this diverse set of electrical features cannot be used when only consumer-focused instrumentation equipments are available. Nevertheless, their work is a valued contribution as it extends the previously mentioned view set out by Norford et al. [NL96] and Laughman et al. [LKC+03].

Further contributing to the development in this area are the features proposed by Wang and Zheng [WZ12]. In addition to the use of steady-state and transient power, basic units of triangles and squares, which are consistent with fast switching events and steady working events respectively, are utilised to represent a power consumption curve across time. Thus, the main outcome of the feature extraction process include geometric properties of these shapes. Interestingly, the concept of power decomposition in this way has also been briefly explored by Cole and Albicki [CA98b] to segment the power consumption curve into slopes and edges.

#### Temporal Features

Temporal features inherent to the operation of appliances have also been utilised in the literature for facilitating load disaggregation. The two main realisations are the dependence of future operational states of appliances on historical states; and the duration of states.

The former, also known as state transition information, is often associated with finite state machine (FSM) appliances. For instance, a washing-machine usually has to visit a series of states sequentially during normal operation; the ”wash” state precedes the ”rinse” state, which in turn leads to the ”spin” state. Being able to take into account the structure of such transitions is one other way in which overlaps in feature space can be resolved. As an example, if there is more than one appliance in the system with power consumption of 150W and we observed a change in steady-state real power value of 150W, we can narrow down the number of potential hypotheses by also considering the state dynamics of competing appliances. While the use of state transition information was first noted by Hart in his seminal paper [Har92], the feature was not actually utilised until much later by Kolter and Johnson [KJ11] and Zia et al. [ZBZ11], among others [PGWR12].

Alternatively, state durations, when being used as signatures, characterise the length of time spent by an appliance in a given operational state. The main idea is that the extent of an appliance being in an ON/OFF state can vary between different appliances and thus, it could be additionally employed to improve disaggregation accuracy. Although the duration is in general dependent on a lot of

other factors (e.g. users' behaviour, time-of-day, day-of-week etc.), well-defined statistical variations can still be modelled and exploited [KAL11]. Refrigerators, for instance, cycle between on and off states nearly periodically, and washing-machines, due to the deterministic control by their internal firmware, spend relatively similar amount of time in the ON state whenever they are being operated. This is in addition to users' habit, which helps to create potentially recurring usage information that is statistically localised. However, in spite of their recent appeal as secondary signatures, only few approaches have employed state durations to date; prominent examples include the work by Kim et al. [KAL11] and the work by Johnson and Willsky [JW13].

Judging from what has been described thus far, it is apparent that temporal features are cost-effective additions to the existing repertoire of standard signatures as they could resolve the feature overlap problem without requiring any underlying changes to existing instrumentation devices. This means, they are highly desirable for situations where the reporting rate of measurements is low (e.g. smart meters) and equipments capable of high sampling rates are not possible to be installed. Though, as we shall see in Section 2.5, there are complexities in modelling them, going by existing approaches [KAL11].

#### Contextual Features

There are many contextual features that could be useful in aiding the process of disaggregation. Appliances which are in close proximity to one another are more likely to be used simultaneously to achieve an intended task (e.g. cooking). The use of air-conditioners is very likely to be correlated with ambient air temperature, and environmental illumination intensity could determine whether lights are being turned on or not. All these are useful auxiliary information that could help to estimate the source of electrical events observed in the aggregate measurements. Therefore, it is natural to see their use in some existing approaches for load disaggregation.

Berges et al. [BSM10], for example, incorporate light, temperature and audio information, together with aggregate real power consumption measurements, to assist the NILM system in distinguishing between appliances which are similar. They set up a wireless sensor network to collect these environmental values in an apartment unit, and it was shown that the events detected in both the auxiliary data and the electrical data are highly correlated with each other. Likewise, Zoha et al. [ZGN+12] followed the same direction but with only one audio sensor. Although the fusion of different kinds of data in this way is interesting,

there is some degree of intrusiveness in the approach. Specifically, installation of additional sensors is needed, with some required to be installed indoors. Moreover, the collection of acoustic information via microphones may not sit well with privacy-conscious users.

On the other hand, Kim et al. [KAL11] included the usage information between different appliances as an additional feature. In their investigation, they noted that the correlation coefficient of the game console with the television is expectedly high, and the use of this appliance dependency information was shown to offer promising improvements in disaggregation accuracy over one without. However, the potential downside to this is the additional overhead in modelling these features when only a very small group of spatially localised appliances are present, not to mention the need for more data to avoid the overfitting problem while the dependencies are learned.

## 2.5 Model Representations of Appliances

Following the extraction of features of interest, the next task is to construct behavioural representations of loads. This involves capturing the linkage between the extracted features and the operational characteristics of appliances, into mathematical models, so that they can be used by algorithms during the disaggregation stage to infer appliance-level contributions. However, because of the potential limits in representing the actual behaviour of appliances, and because of possible assumptions made in the representation to ease computational requirements, the choice of models used also greatly affects the accuracy of disaggregation. Therefore, this section describes the state-of-the-art appliance models which have been employed in existing approaches, in terms of two main categories: generative models and discriminative models, before discussing how model parameters are generally learned in practice.

### 2.5.1 Generative Models

In statistics and machine learning, a generative model includes in its specification the details by which the observed variable is generated from a hidden variable. Thus, it describes how, given a specific realisation of the latter, a sample of the former can be produced. In the case of NILM, the measured aggregate values and the extracted features are examples of observed variables, while the operational

states of appliances and the underlying appliance-specific measurements can be considered hidden variables.

Suppose we have an observed variable $y$ and a hidden variable $x$. The generative model can simply be represented by the conditional probability $p(y \mid x)$. From Hart's technical report of his prototype [Har85], it is this form that has been used. In particular, the pair $(\Delta P, \Delta Q)$ is treated as $y$ and a bivariate Gaussian distribution is employed for characterising each cluster/appliance $x$ in the $\Delta P$-$\Delta Q$ plane. Though, one issue with his representation is the use of only two states: ON and OFF. Provisions for modelling multi-state appliances are not integrated into his prototype.

Also related is the use of Gaussian mixture model (GMM) by Chou and Chang [CC13], where the operational states of a group of appliances are represented by mixtures in the real power domain and each mixture corresponds to a particular state. While a natural representation, GMM as used in these approaches assumes that $x$ at different times are independent, whereas in reality, this is not true, given that $x$ at time $t$, $x_t$, is very likely to be correlated with $x_{t-1}$.

For these reasons, representations based on hidden Markov models (HMM) have been proposed by subsequent work [KJ11, ZBZ11, PGWR12]. Similar to a GMM, the conditional probability $p(y \mid x)$ is still maintained, but, now, state transition information like those described in Section 2.4.3 are incorporated. The outcome is a probabilistic generative model that expresses the relationship between a sequence of $T$ hidden states, $x_{1:T}$, and a sequence of $T$ observed variables, $y_{1:T}$, where the subscript $1:T$ denotes the time index from 1 to $T$, i.e. $y_{1:T} = (y_1, \ldots, y_T)$.

The model assumes that the current state, $x_t$, is only dependent on the previous state, $x_{t-1}$, i.e. $p(x_t \mid x_{1:t-1}) = p(x_t \mid x_{t-1})$, while the observed value at time $t$, $y_t$, is considered to be "emitted" based on the value of the hidden state at the same time step. As such, the model is characterised by the conditional probability $p(y_{1:T} \mid x_{1:T})$ whose corresponding joint probability is

$$\begin{aligned} p(x_{1:T}, y_{1:T}) &= p(x_{1:T}) p(y_{1:T} \mid x_{1:T}) \\ &= p(x_1) \textstyle\prod_{r=2}^{T} p(x_r \mid x_{r-1}) \prod_{s=1}^{T} p(y_s \mid x_s), \end{aligned} \tag{2.5}$$

where $p(x_\tau \mid x_{\tau-1})$ is the state transition probability and $p(y_s \mid x_s)$ is the emission probability. The model parameters $\boldsymbol{\lambda}$ governing an HMM consist of the initial state probability $\pi$, the state transition probability matrix $A$ and the parameters of the emission probability $B$. The element of the $i$th row and the $j$th column in matrix $A$, $a_{ij}$, is the probability of transitioning from state $i$ to state $j$. On the other hand, the parameters of the emission probability $B$ depend on the distribution

of $p(y_s \mid x_s)$. If $p(y_s \mid x_s)$ is a Gaussian distribution, then $B$ contains the mean and variance of $y_s$ for a given $x_s$. Shown in Figure 2.5 is the dynamic Bayesian network (DBN) of HMM, with square nodes denoting discrete random variables and round nodes symbolising continuous random variables. Shaded nodes on the other hand indicate that the variables are given or observed while arrows signify the conditional dependence relationship between nodes.

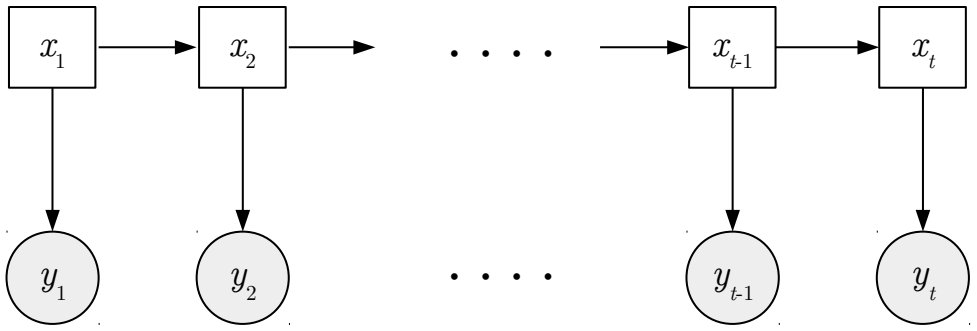



Figure 2.5: Dynamic Bayesian network of HMM.

For modelling appliance behaviour, $y_t$ is the power consumption at time $t$ and $x_t$ is the internal state of appliance at time $t$. Specifically, for the disaggregation problem, $y_t$ and $x_t$ could refer to the aggregate power consumption and the system state at time $t$ respectively, where the latter is the combined state of a group of appliances [MPB+16]. That is, in a hypothetical house with two binary-state appliances, $x_t = a$ can refer to a fan being ON and a light being OFF, and $x_t = b$ can signify both being OFF. Though, in this situation, it is preferable to name the system states by vectors (e.g. $\mathbf{x}_t$), denoting the state of each appliance in that system state.

To that end, factorial hidden Markov model (FHMM), an extension of HMM, has been employed in the literature [KJ12, EBE15]. It retains the properties of an ordinary HMM, except that, there are now $K$ independent chains, each contributing to the observed aggregate measurements, $y_{1:T}$. Therefore, like the HMM, the DBN of FHMM can be visualised as that of Figure 2.6; and the joint probability of the system states, $\mathbf{x}_{1:T}$, and the observed variables, $y_{1:T}$, is

$$p(\mathbf{x}_{1:T}, y_{1:T}) = p(\mathbf{x}_1) \prod_{r=2}^{T} p(\mathbf{x}_r \mid \mathbf{x}_{r-1}) \prod_{s=1}^{T} p(y_s \mid \mathbf{x}_s), \tag{2.6}$$

where $p(\mathbf{x}_1) = \prod_{k=1}^{K} p(x_{1,k})$, $p(\mathbf{x}_r \mid \mathbf{x}_{r-1}) = \prod_{k=1}^{K} p(x_{r,k} \mid x_{r-1,k})$ and $x_{t,k}$ refers to the internal state of appliance $k$ at time $t$.

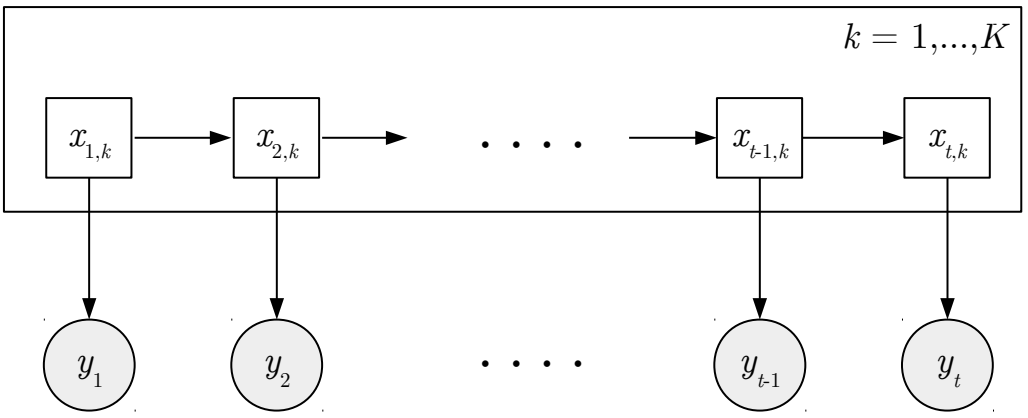



Figure 2.6: Dynamic Bayesian network of FHMM.

Although HMM-based models have been shown to work well in general, the Markov assumption still limits the potential for disaggregation, since it implicitly restricts the state durations of appliances to geometric/exponential distributions [Yu10]. This means, especially for cases where appliances have largely characteristic state durations, useful information which could otherwise help to resolve feature overlaps is disregarded. In light of this, Kim et al. [KAL11] proposed the use of a model based on hidden semi-Markov model (HSMM) for representing durations of the ON state explicitly, coupled with additional means to account for extra features from auxiliary data and dependencies between appliances. However, despite being one of the most flexible models proposed to date, their model is limited to describing binary-state appliances. Therefore, apart from its high computational requirements, it may have issues with modelling behaviours of multi-state loads.

In a similar vein, Johnson and Willsky [JW13] introduced the hierarchical Dirichlet process hidden semi-Markov model (HDP-HSMM) for Bayesian modelling of power consumption values and state durations but without restrictions to binary-state appliances; and in agreement with Kim at al. [KAL11], it was shown that HSMM-based approaches do indeed outperform HMM-based models. Up to this point however, all HSMM-based models used for NILM have been instances of the explicit-duration formulation. As such, calculations of probability values at any given time step, needed for real-time inference of states, are difficult (see Chapter 3 for more details). Also, hard-coded bounds on the duration space generally have to be made during inferences of hidden states, unless a specific non-parametric approach is used [DWW12].

Other non-probabilistic generative models which have been used for NILM are sparse coding [KBN10] and matrix factorisation techniques such as non-

negative matrix factorisation [FRA13], given their underlying generative interpretation [LS99].

### 2.5.2 Discriminative Models

Discriminative models form one other type of representation for associating observed variables with hidden variables. However, unlike generative models, which model the statistical variation of the observed variables with respect to that of the hidden variables, discriminative models only care about the unidirectional mapping from the former to the latter. In other words, discriminative modelling involves constructing a function whose inputs are observed variables and outputs are hidden variables. For example, given an observed variable $y$ and a hidden variable $x$, the task could be finding the form of $p(x \mid y)$ directly based on a set of training data (e.g. labelled pairs of $(x, y)$). Therefore, in contrast to generative models, no assumptions on the distribution of $p(y \mid x)$ (e.g. Gaussian distribution) needs to be made [Jor95], as the one-way mapping is learned without having to go through the Bayes relation, $p(x \mid y) = p(y \mid x)p(x)/p(y)$. Among the common realisations used in NILM are artificial neural networks (ANN), support vector machines (SVM) and $k$-nearest neighbours ($k$-NN).

In the work of Srinivasan et al. [SNL06], a vector denoting the harmonics of the aggregate current signal is the observed variable, while another binary vector describing the presence of appliances in the aggregate measurements is the hidden variable. The association between the two variables is then represented using a 3-layer ANN. Similarly, Ruzzelli et al. [RNSO10]; and Chang and Lee [CL13] both employed a 3-layer ANN for load identification. However, experiments conducted for each of these approaches are mostly exploratory in nature, with limited tests conducted to quantify their applicability in real houses with many appliances. Moreover, the feedforward architecture of the ANN used in these cases are generally incapable of modelling the temporal dependencies between extracted features of different times [SNS15], thus preventing the sequential structure of the electrical data from being used to improve disaggregation.

Also related is the work by Kelly and Knottenbelt [KK15a]. They provided extensions to previous approaches by adapting deeper variants of ANN for NILM. Three different deep neural network architectures with up to 8 layers were investigated, and it was found that the long short-term memory (LSTM) architecture performs the best overall in detecting two-state appliances. Though, it was noted by the authors that further investigations need to be done to identify why

the model does not work well with multi-state appliances. Other apparent disadvantages are the need for lots of training data to fit models with millions of parameters and the relatively high computational requirements during the training stage. The paper reported that, between 1 and 12 hours are required to train each network on a Graphics Processing Unit (GPU).

SVM has also been considered in NILM. The main idea is to divide the feature space into regions formed by boundaries, with each region corresponding to a group of inputs with the same output. Given a set of training data, the role of training a SVM is then to find these boundaries. Patel et al. [PRK$^+$07] and a related follow-up work by Froehlich et al. [FLC$^+$09], for example, utilised SVM to map high-dimensional vectors consisting of frequency components of a transient voltage noise to contributing appliances. Likewise, a number of other approaches with spectrum-based signatures have also characterised the behaviour of loads in this way [LWP12, JLL$^+$12].

Lastly, $k$-NN, being a non-parametric method, does not build explicit representations of features. It simply uses labelled training data as it is, and constructs a function that assigns a newly observed variable, $y_{\text{new}}$, to the class with the largest membership count amongst $k$ of $y_{\text{new}}$'s closest points in the training data. For example, with $k = 3$, the 3 closest points to $y_{\text{new}}$ (e.g. in terms of the Euclidean distance) are considered. If 2 of those points belong to the class C1 and the remaining one point is labelled with class C2, then $y_{\text{new}}$ is assigned to class C1 by virtue of being the majority. Due to its simplicity, 1-NN has been employed in the work of Gupta et al. [GRP10] and the work of Berges et al. [BGMS10] for mapping features vectors to the contributing appliances.

Overall, while it has been considered in the field of machine learning that discriminative models have generally lower error rates than that of generative models in classification tasks, their performance can be relatively poor when the size of the training set is small [NJ02]. For this reason and the observation that datasets for NILM are still limited in quantity at the time of this research, discriminative models are not used as the basis for our proposed model. Though, as a proposal, its inclusion in the form of a hybrid is easily realisable without changing the fundamental concept of our approach (see Chapter 6), should there be more data in the future.

### 2.5.3 Learning

One important concern when building appliance models is the way in which their parameters are learned. For probabilistic generative models, this includes distributional parameters like mean and variance, whereas for SVM, it refers to the boundaries. Depending on whether or not a labelled dataset is present, the parameters could either be learned automatically via unsupervised means or manually with the help of labelled data.

The latter, also known as supervised learning, requires appliance-level data corresponding to a particular setting to be made available. Learning in this way involves a training phase, whereby model parameters governing the relationship between the aggregate-level features and the true states of appliances is established. This a process which is very much analogous to finding the parameters of a linear equation given a set of values for the target variable and the independent variable.

On the other hand, unsupervised learning does not depend on the availability of appliance-level data and learning has to be performed through aggregate-level features only. While this is more practical in that the one-time collection of appliance-specific data can be avoided, it is a more challenging problem. For example, unsupervised learning can be an ill-defined problem; besides having to contend with the issue of feature overlaps, it also has to account for the possibility that the surface of the objective function over the model space is non-convex with multiple modes. As such, supervised learning remains one of the more mainstream approaches for NILM to date. A more concrete description of the different learning approaches are presented as follows.

#### Supervised Learning

We have seen the various representations that could be used to model the relationship between the observed variables and the hidden variables in Section 2.5.1 and Section 2.5.2. To actually learn the specific forms of representations however, we need a training set with $T$ labelled data points, i.e. $\{(x_t, y_t)\}_{t=1}^{T}$, of which $x_t$ and $y_t$ refer to the $t$th label and the $t$th signature value respectively. Especially for discriminative models, this means finding the parameters of a function, $f: Y \rightarrow X$, such that for all $t$, $x_t \in X, y_t \in Y$, whose form is governed by the representation used.

In the case of ANNs, gradient descent via the backpropagation technique is the standard means of learning the parameters (e.g. the weights of the inter-

neuron connections), whereas for SVM, the parameters of the boundary equation are obtained by solving a convex optimisation problem using constrained quadratic programming. For a more detailed background on these, see [Kot07].

### Unsupervised Learning

Within the NILM literature, unsupervised learning has been the primary tool used for generative models [KAL11, CC13]. Given only the aggregate measurements, i.e. $\{(y_t)_{t=1}^T\}$, the task is to find the parameters governing the distribution $p(x_{1:T}, y_{1:T})$ even when the labels $x_{1:T}$ are not given.

The standard technique for achieving this is an iterative method, known as the Expectation-Maximisation (EM) algorithm. Depending on the representations imposed on $y_{1:T}$ (e.g. GMM *vs* HMM etc.), the specific formulation of EM can differ. However, the general description of EM remains the same in each case. Therefore, EM is described in this manner as part of the brief exposition that follows. A more complete description of EM can be found in [Bil98].

Suppose that $\boldsymbol{\lambda}$ denotes the parameters of the distribution, then the natural goal for estimating $\boldsymbol{\lambda}$ is to maximise the likelihood function

$$l(\boldsymbol{\lambda} \mid x_{1:T}, y_{1:T}) = p(x_{1:T}, y_{1:T} \mid \boldsymbol{\lambda}) \tag{2.7}$$

with respect to $\boldsymbol{\lambda}$. But, because the labels or hidden states $x_{1:T}$ are also unknown and the aggregate measurements $y_{1:T}$ are the only constants, it is not possible to perform the maximisation directly. For that, an initial guess for $\boldsymbol{\lambda}$, $\boldsymbol{\lambda}'$, has to be assumed. Then, using $\boldsymbol{\lambda}'$ as the starting point, the estimate of $\boldsymbol{\lambda}$ is iteratively updated.

There are two steps in one iteration: Expectation (E-step) and Maximisation (M-step). The E-step involves computing the expectation

$$\mathcal{Q}(\boldsymbol{\lambda}, \boldsymbol{\lambda}^{[i-1]}) = E\left[\log(p(x_{1:T}, y_{1:T} \mid \boldsymbol{\lambda})) \;\middle|\; y_{1:T}, \boldsymbol{\lambda}^{[i-1]}\right] \tag{2.8}$$

in terms of $\boldsymbol{\lambda}$, given the previous iteration's estimate, $\boldsymbol{\lambda}^{[i-1]}$. Then, in the M-step, the maximisation

$$\boldsymbol{\lambda}^{[i]} = \arg\max_{\boldsymbol{\lambda}} \mathcal{Q}(\boldsymbol{\lambda}, \boldsymbol{\lambda}^{[i-1]}) \tag{2.9}$$

is performed. Both the E-step and the M-step are repeated until the likelihood function corresponding to the estimated $\boldsymbol{\lambda}$ converges. While EM's convergence is guaranteed and its properties are well-known [Wu83], it only performs a local

search. Therefore, there is no guarantee that the estimate of $\boldsymbol{\lambda}$ is globally optimum. One of the common workarounds for this is to perform multiple runs of EM, with randomly sampled initial guess for each round. The final estimate is then chosen based on the run that gives the largest likelihood value. The description of other workarounds are given in [KX03].

Computationally, the EM algorithm is tractable for less complex models like GMM and HMM. However, this is not true for the more flexible variants like FHMM and FHSMM as used in NILM [KAL11]. In particular, the E-step under such models are challenging to compute and thus, sampling methods such as Gibbs sampling have been employed to approximate the expectation in (2.8) [GJ97].

One other issue is that, despite being unsupervised, EM still requires the state space to be known a priori. In the case of FHMM, this means the number of states for each of the $K$ chains in Figure 2.6 needs to be specified. As this is typically not known in advance (e.g. the number of states per appliance and the number of appliances), the use of EM may be potentially limited from a practical point of view. To that end, Johnson and Willsky [JW13], in their unsupervised approach, proposed the use of a Bayesian non-parametric method of learning the number of states inherent in the aggregate measurements.

#### Semi-supervised Learning

Besides the two groups of learning paradigm mentioned previously, there exists a relatively new class of approaches which is used in NILM: semi-supervised learning. Though considered unsupervised from the perspective of end-users, the work of Parson et al. [PGWR14] actually combines both supervised and unsupervised learning. The latter comes from the fact that appliance-level data from a large number of appliances of different brands is utilised to create generic appliance models. Then, during the roll-out stage, these models are tuned in an unsupervised manner to house-specific appliance models using Bayesian inference.

While no labelled data is directly needed as far as the user is concerned, the creation of generic appliance models still necessitates a deep understanding of each class of appliances considered. For example, the designer of a particular NILM implementation might have to perform a large scale collection of data to investigate generalisable properties.

Moreover, it is not entirely clear whether classes of appliances apart from those studied by Parson et al. [PGWR14] can be encoded into a general form. Some ap-

pliances in particular are especially tied to the behaviours or habits of users (e.g. television). In this regard, it might be non-trivial to deduce a generic property that could apply to a wide number of cases.

## 2.6 Disaggregation

Given the appliance models and the learned model parameters, the final task is to decode the observed aggregate measurements into appliance-wise contributions. Subject to the types of models used, there are a number of ways in which this could be achieved. In this section, we present two main groups of approaches generally used in the literature: optimisation methods and machine-learning techniques as applied to latent variable models.

### 2.6.1 Optimisation Methods

Load disaggregation is intrinsically a combinatorial optimisation problem; the aggregate measurements are summations of the appliance-specific measurements whose exact values are unknown and the size of the solution space grows exponentially in the number of appliances. If we suppose there are $T$ sequential samples of aggregate values, $y_{1:T}$, and $K$ appliances, then the problem of estimating each of the $K$ appliances' contributions can be formulated as

$$(\hat{y}_{1:T}^{(1)}, \ldots, \hat{y}_{1:T}^{(K)}) = \underset{(y_{1:T}^{(1)}, \ldots, y_{1:T}^{(K)})}{\arg\min} \left\| y_{1:T} - \sum_{k=1}^{K} y_{1:T}^{(k)} \right\|, \tag{2.10}$$

where $y_{1:T}^{(k)}$ refers to the unknown measurements of appliance $k$ and $\|\cdot\|$ can be any appropriate norm (e.g. $\ell_1$ or $\ell_2$ etc.).

Instances of this in existing approaches are the work of Suzuki et al. [SIS+08] and more recently, the work of Kong et al. [KDH+16], both of which employed methods based on integer programming. In acknowledging the computational intractability inherent to performing optimisation directly, the latter included various heuristics specific to the problem of disaggregation to reduce time complexity. However, like most methods based on combinatorial optimisation, it has issues in performing well when unknown appliances are present, as it assumes all $K$ appliances can be modelled and the aggregate values necessarily include the contributions of only these appliances in question.

Though, one prominent exception to this is the work by Kolter and Jaakkola [KJ12]. Being a reformulation of FHMM into an equivalent convex optimisation problem, their approach specifically included additional variables and regularisation based on compressive sensing to account for the existence of unmodelled or unknown loads in the aggregate signal. Therefore, it is one of the most notable approaches in this aspect. That said, because the optimisation procedure is not an incremental algorithm, disaggregation has to be done in blocks of aggregate measurements, violating real-time requirements.

### 2.6.2 Inference of Hidden Variables

In latent variable models like those discussed primarily in Section 2.5.1 and Section 2.5.2, the main objective during disaggregation is to infer the hidden variables $\mathbf{x}_{1:T}$ (e.g. the operational states of all appliances) given the observed variables $y_{1:T}$ (e.g. aggregate measurements/features). For those based on discriminative models, this entails a simple straightforward computation in which the learned function is used to map any newly observed values to estimates of hidden variables, whereas it is a more involved process in the case of generative models, especially probabilistic ones. As such, we will limit our discussion in this subsection to the latter.

Assuming that the model parameters $\boldsymbol{\lambda}$ have been learned, the task of estimating $\mathbf{x}_{1:T}$ given $y_{1:T}$ is closely related to the posterior probability of $\mathbf{x}_{1:T}$, i.e. $p(\mathbf{x}_{1:T} \mid y_{1:T})$. For example, a natural thing to do is to find the $\mathbf{x}_{1:T}$ that maximises $p(\mathbf{x}_{1:T} \mid y_{1:T})$ or $p(\mathbf{x}_{1:T}, y_{1:T})$, a procedure also known as maximum a posteriori probability (MAP) estimation. Thus, the problem can be formally formulated as

$$\hat{\mathbf{x}}_{1:T} = \arg\max_{\mathbf{x}_{1:T}} p(\mathbf{x}_{1:T}, y_{1:T}). \tag{2.11}$$

However, because the maximisation of (2.11) is computationally intractable, it is not performed directly. Instead, a number of standard methods such as the Viterbi algorithm, particle filter, Gibbs sampling and heuristic methods are employed for state inference in NILM. A brief description of these methods are given below.

#### Viterbi algorithm

The Viterbi algorithm is frequently used as part of state inference under HMM-based models. For a basic HMM with a single chain $x_{1:T}$, finding the most likely state sequence is achieved by using a dynamic programming paradigm and the

Markov property. This enables the recursive expression

$$\delta_t(j) = \begin{cases} p(x_1 = j)p(y_1 \mid x_1 = j), & \text{if } t = 1 \\ \max\limits_i \delta_{t-1}(i)p(x_t = j \mid x_{t-1} = i)p(y_t \mid x_t = j), & \text{otherwise} \end{cases} \tag{2.12}$$

to be derived.

The Viterbi algorithm then begins by computing and storing the Viterbi score $\delta_t(j)$ at each time step $t$ and for each state $j$, while the selected state for the maximisation over $i$ in (2.12) is recorded using a backpointer

$$\psi_t(j) = \arg\max_i \delta_{t-1}(i)p(x_t = j \mid x_{t-1} = i)p(y_t \mid x_t = j).$$

After the last measurement $y_T$ is observed, a backtracking procedure is performed to obtain the estimate $\hat{x}_{1:T}$ such that

$$\hat{x}_T \leftarrow \arg\max_j \delta_T(j)$$
$$\hat{x}_t \leftarrow \psi_{t+1}(j) \text{ for } t = T - 1 \text{ to } 1.$$

Computationally, if there are $M$ possible states that $x_t$ can hold, the time complexity of the maximisation operation with the Viterbi algorithm is $O(M^2T)$. Although this is a significant improvement over the direct naive attempt of (2.11) with $O(T^M)$, state inference using the Viterbi algorithm under FHMM or the FHMM-equivalent HMM is still not computationally tractable in practice [GJ97]. For instance, in the case with $K$ 2-state appliances (i.e. $K$ chains and $M$ = 2), the time complexity is $O(2^{2K}T)$, that is, the number of required computations grows exponentially in the number of appliances. Nevertheless, by using various assumptions to sidestep intractability, modifications based on the original Viterbi algorithm have been utilised in NILM as follows.

In the work of Zeifman and Roth [ZR11b], only a subset of two appliances, whose distributions of change-in-power value are close with one another, are considered at a time for detection using the Viterbi algorithm. The assumption is that, other distributions corresponding to other appliances are sufficiently far away such that their removal from consideration in the detection of the two appliances in question is justified. This means, for each partially overlapping subset to be processed independently by the Viterbi algorithm, the number of states involved is reduced significantly. However, despite being an interesting development, the study only accounts for binary-state appliances. Further, the approach

may have trouble handling observed change-in-power values which are actually caused by any of the two appliances but fall beyond the bounds for which the likelihoods under the corresponding distributions are insignificant. This is on top of complications in dealing with change-in-power values attributable to two or more appliances.

Also related is the more recent work by Makonin et al. [MPB+16]. In their approach, the naturally sparse state transition matrix of HMM, when used in NILM, is exploited to avoid unnecessary computations and to reduce the memory cost of the Viterbi algorithm. While it has been demonstrated by the authors that the developed technique is able perform really well, there are several concerns. Firstly, the modified Viterbi algorithm is no longer in essence a Viterbi algorithm, since it does not estimate the most likely state sequence. Rather, for every pair of observed values (i.e. $y_{t-1}$ and $y_t$), it infers $x_t$ as $\hat{x}_t = \arg\max_j \delta_t(j)$. As this is a greedy approach, it may prematurely discard correct solutions which have low Viterbi scores given only measurements observed thus far but have high Viterbi scores in light of future observations. Therefore, the state sequence inferred in this way might be suboptimal. Secondly, the problem of performing disaggregation in the presence of unmodelled loads is not specifically addressed. As such, the introduction of new appliances via new purchases or guest visits may negatively affect disaggregation accuracy.

#### Particle filter

The particle filter (PF), also known as Sequential Monte Carlo (SMC), offers an alternative approach to estimating the underlying state sequence by allowing samples of state sequences to be obtained via a recursive sampling process [DJ08]. In this way, the posterior distribution $p(x_{1:T} \mid y_{1:T})$ as used in Bayesian inference can be approximated numerically in real-time even in cases where the state $x_t$ is high-dimensional and the posterior distribution cannot be expressed in tractable form.

The main components of particle filters are a proposal distribution $q(x_{1:T})$ and a weight function $w(x_{1:T})$. Suppose that the posterior distribution $p(x_{1:T} \mid y_{1:T})$ is desired but cannot be expressed in exact form, and suppose that the direct sampling of $x_{1:T}$ from $p(x_{1:T} \mid y_{1:T})$ is not possible. If we can define an alternative distribution (i.e. proposal distribution) that we can easily sample from such that $q(x_{1:T}) > 0$ when $p(x_{1:T} \mid y_{1:T}) > 0$ for all $x_{1:T}$, then the posterior distribution can

be approximated by first drawing $N_\text{p}$ samples from $q(x_{1:T})$ before evaluating

$$w(x_{1:T}^{(i)}) = \frac{p(x_{1:T}^{(i)}, y_{1:T})}{q(x_{1:T}^{(i)})} \tag{2.13}$$

$$\hat{p}(x_{1:T} \mid y_{1:T}) = \sum_{i=1}^{N_p} w(x_{1:T}^{(i)}) \delta_{x_{1:T}, x_{1:T}^{(i)}}, \tag{2.14}$$

where $\hat{p}(x_{1:T} \mid y_{1:T})$ is the approximated version of $p(x_{1:T} \mid y_{1:T})$, $x_{1:T}^{(i)}$ is the $i$th sample drawn from $q(x_{1:T})$ and $\delta$ is the Kronecker delta function.

However, as particle filters are described as a sequential process in which $N_\text{p}$ samples are not drawn as blocks of $T$ states, but instead, sampled incrementally at each time step $t$, a recursive form for $q(x_{1:t})$ and $w(x_{1:t})$ have to be defined. In general, both are given as

$$\begin{aligned} q(x_{1:t}) &= q(x_{1:t-1}) q(x_t \mid x_{1:t-1}) \\ &= q(x_1) \prod_{\tau=2}^{t} q(x_\tau \mid x_{1:\tau-1}) \end{aligned} \tag{2.15}$$

$$\begin{aligned} w(x_{1:t}) &= w(x_{1:t-1}) \alpha(x_{1:t}) \\ &= w(x_1) \prod_{\tau=2}^{t} \alpha(x_{1:\tau}), \end{aligned} \tag{2.16}$$

where $q(x_t \mid x_{1:t-1})$ and $\alpha(x_{1:t})$ refer to the incremental proposal distribution and the incremental weight function respectively.

With that, the $i$th sample up to time $t$, $x_{1:t}^{(i)}$, can be obtained by drawing $x_t^{(i)}$ from $q(x_t \mid x_{1:t-1})$, having already drawn $x_{1:t-1}^{(i)}$ in previous time steps. In the same way, weights of the previous time steps are used to calculate the current weights incrementally. Lastly, to mitigate the sample degeneracy problem inherent in the incremental sampling process, whereby the weights of samples reduced to significantly low values after a number of time steps, a resampling procedure is performed. This involves drawing $N_p$ new particles with replacement from the pool of $N_\text{p}$ particles (i.e. $\{x_{1:t}^{(i)}\}_{i=1}^{N_\text{p}}$), with the probability of selecting each $i$th particle being the normalised weight function

$$\tilde{w}(x_{1:t}^{(i)}) = \frac{w(x_{1:t}^{(i)})}{\sum_{j=1}^{N_\text{p}} w(x_{1:t}^{(j)})}, \tag{2.17}$$

whenever the effective sample size (ESS)

$$\text{ESS} = \left( \sum_{i=1}^{N_\text{p}} w(x_{1:t})^2 \right)^{-1} \tag{2.18}$$

falls below a predefined threshold. Theoretically, the resampled particles should reflect the posterior distribution $p(x_{1:t} \mid y_{1:t})$.

In NILM, particle filters have been predominantly used by Egarter et al. [EBE13, EBE15] to infer contributions of appliance power consumption under FHMM. They showed that appliance models would not need to be learned precisely in order to disaggregate properly. Though, as admitted by the authors, the approach is not able to perform well when appliances with similar power consumption are present. Furthermore, underlying the resampling step of particle filters is the issue of low particle diversity, as particles with high weights are drawn more often, potentially taking more spots in the pool of $N_\text{p}$ resampled particles at the expense of particles with low weights. In the extreme case, these particles would all be represented by only a single realisation of $x_{1:t}$. In this regard, the issue is especially wasteful for NILM, considering that an observed aggregate signal could be explained by many different combinations of component signals, and the loss in diversity would prematurely lock-in estimates of state sequence before more future measurements are observed. Even if on-demand resampling based on the aforementioned ESS is employed, there is no guarantee that uniqueness amongst resampled particles are maintained. Not to mention, it is difficult to ascertain the ESS threshold to use in advance.

#### Gibbs sampling and heuristic methods

Similar to particle filters, the Gibbs sampler is a Monte Carlo method for approximating multivariate distributions, but samples are instead drawn iteratively from a Markov Chain, which composed of conditional probabilities of the multivariate random variables in question, and whose equilibrium distribution mirrors that of the target distribution. Examples as used in NILM are the earlier work of Kolter and Johnson [KJ11] for performing inference under FHMM, and the work of Johnson and Willsky [JW13] for jointly inferring the states of appliances and the parameters of their proposed HDP-HSMM model mentioned in an earlier section.

Apart from sampling methods, heuristics for solving optimisation problems have also been used. In particular, Kim et al. [KAL11] utilised simulated annealing (SA) for performing maximum likelihood estimation on the state sequence,

given that the exact state inference using the Viterbi algorithm under his proposed HSMM-based model is not computationally tractable.

While the contribution of the approaches mentioned here is that they use more complex appliance models, the use of both Gibbs sampling and SA means they are only inherently suited for batch processing; blocks of measurements have to be observed (e.g. a day's worth) before retrospective inference of states is performed. As such, these techniques do not lend themselves well to meeting one of our research objectives, that is, the real-time disaggregation of power consumption measurements.

## 2.7 Limitations: A Summary

We have provided a detailed overview of the various aspects of NILM methods as used in the literature in previous sections, while noting the drawbacks and benefits of each category. In this section, we summarise the limitations to give a clearer perspective on unsolved problems in the field to motivate the need for our proposed approach as described in the chapters that follow.

Firstly, the common assumption of one-at-a-time (i.e. only one device per transition) as alluded in Section 2.3.2 does not normally hold for low sampling rates. While it is a reasonable assumption in reducing computational complexity in high sampling rates application, restricting one appliance to change state at any given time might cause the algorithm to be less robust, on top of being more likely to induce errors in low sampling rate scenarios. The observation that several recent works [DROS13, KJ12, RCG12, SY12, KZZS13] still employed this assumption means that more attention needs to be devoted for relaxing this constraint without affecting computational tractability.

Secondly, as far as generative models are concerned, the use of factorial hidden Markov models (FHMM) in few of the recent, prominent works [KJ12, EBE15, MPB$^+$16] implicitly assumes that the state duration is geometrically distributed. Essentially, this Markov property means the probability of transitioning to a new state is only dependent on the previous state, regardless of the actual duration in which a given appliance has been in its previous state for. Even without accounting for the recurring habits of household occupants, the assumption is limiting as a number of common appliances (e.g. refrigerators, water pump etc.) are cyclic in nature. Moreover, the assumption prevents useful patterns expressible via state duration information to be leveraged, severely restricting the potential of resolving overlaps in feature space.

Thirdly, although state-of-the-art methods utilising state durations have been explored in the form of hidden semi-Markov models (HSMM) to address the second limitation [KAL11, JW13, GWK15], the variant used does not inherently allow real-time processing. For instance, the duration random variable as expressed in these works require blocks of data to be obtained before the probability can be computed, whereas in a real-time system, it is expected that calculations could be updated incrementally as new measurements are observed. In addition, for state inference under these models, hard-coded bounds on the duration space have to be made typically [DWW12], thus reducing robustness.

Last but not least, it is widely believed that all appliances in a house can be accounted for during the training/learning stage. Methods for dealing with unknown appliances have been few and far between. A thorough search in the literature only yielded one work by Kolter and Jaakkola [KJ12], and another more recently by Tang et al. [TWLT16], whereby a robust mixture residual term has been introduced to explicitly take on contributions from unknown appliances. The lack of investigation in this aspect largely inhibits the wide adoption of NILM in the real world. Furthermore, the two approaches mentioned are inherently non-real-time, as they perform disaggregation in batches.

## 2.8 Public Datasets

Aside from the direct problem of disaggregation, one of the main issues in the literature is the use of non-standard private datasets for the evaluation of NILM approaches. Not only does this prevent claimed disaggregation accuracies in past papers to be validated, it also makes disaggregation accuracies not comparable across a broad set of work. It is only until recently that the uptake of common datasets increases. While the number of public datasets is still relatively limited, it is gradually growing to facilitate more research in load disaggregation. In this section, a brief overview of the common public datasets is provided, in addition to remarks on their relevance to our research scope.

### 2.8.1 Reference Energy Disaggregation Dataset (REDD)

The REDD dataset is the first public dataset specially catering to the development of NILM algorithms. Since its release by Kolter and Johnson [KJ11] in 2011, the number of proposed NILM algorithms tested against REDD has grown. Indi-

rectly, this marks the start of the use of common datasets in the literature as before that, NILM algorithms have only been validated against private datasets.

The dataset consists of appliance-level data and aggregate-level measurements in the form of real power quantities and apparent power quantities[3] respectively. The data are collected from 6 houses in the Greater Boston area in the United States, with monitoring durations up to nearly 2 months. Also made available are voltage waveforms and current waveforms sampled with a rate of 16.5kHz from the main metering point of two houses.

The REDD dataset is considered as part of our evaluation as it is widely regarded by the NILM community as the standard dataset for benchmarking NILM algorithms. Apart from that, it is also relevant for testing our proposed method due to its inclusion of ground truth appliance power consumption data with sampling intervals in the order of seconds. The voltage waveforms and current waveforms in the dataset are however not used, given our research focus on low frequency data obtainable from typical smart meters.

### 2.8.2 Building-Level Fully-Labeled Dataset for Electricity Disaggregation (BLUED)

After Kolter and Johnson started the trend of publishing datasets for NILM, Anderson et al. [AOB+12] released the BLUED dataset in 2012, with specific emphasis on motivating research of event-based approaches, whereby detectable events in the aggregate signal are individually classified. To that end, they made available appliance-level ground truth data in the form of events corresponding to the actual state transition of appliances in a single family house in Pittsburgh, United States. Accompanying that data is also the aggregate-level voltage and current measurements sampled at a rate of 12kHz. Duration-wise, the monitoring spans a week.

While the dataset might be useful in some aspects, its use is not considered in our evaluation primarily because of the lack of appliance-level power consumption data. Moreover, its emphasis on validating event-based approaches is orthogonal with our research scope and modelling approach in Chapter 3.

[3] Although it was not mentioned by Kolter and Johnson [KJ11], there is a consensus that the aggregate-level measurements are actually apparent power quantities due to the scaling of the aggregate-level measurements relative to the what was observed in the appliance-level data [BDS13, Zei12].

### 2.8.3 Almanac of Minutely Power Dataset (AMPds)

Makonin et al. [MPB+13] follow suit with the publication of the AMPds dataset in 2013. The dataset consists of appliance-level and aggregate-level real power measurements, reactive power measurements, root mean square (RMS) current and voltage data, on top of aggregate-level gas flow measurements and water flow measurements, each obtained at every minute. Data collection is performed in a house in the Greater Vancouver region of British Columbia, Canada for a duration of one year. The availability of non-electrical data is envisioned to be useful for NILM algorithms that employ external features.

As only one house is instrumented, the dataset is not chosen for our evaluation. Though, for future work, it will be useful for modelling seasonal variation in appliance behaviour given the long monitoring duration of one year.

### 2.8.4 UK Domestic Appliance-Level Electricity (UK-DALE)

The UK-DALE dataset was released by Kelly and Knottenbelt [KK15b] in 2015. It is the most comprehensive dataset to date, comprising of data measured from 5 houses in the United Kingdom, with monitoring periods of up to 2 years and 54 channels' worth of appliance-level data for one of the houses. Appliance-level measurements and aggregate-level measurements are made available in real power and apparent power quantities obtained at sampling intervals in the order of seconds, besides the aggregate-level voltage and current waveforms, each sampled at a rate of 16kHz.

The breadth and depth of the data collected are certainly beneficial for a more detailed appliance behavioural analysis and modelling. Unfortunately, at the time of the development of our proposed method, the dataset did not exist. Therefore, the data can only be considered as part of any future work for investigating the aforementioned seasonal variation and the enlarged scale of appliance classes.

## 2.9 Research Scope

With this background, it is now possible to elaborate on the research goals listed in Chapter 1.

First, the focus is solely on disaggregating aggregate-level real power measurements with sampling intervals in the order of seconds up to a few minutes. It is assumed that no other power quantities like reactive power is available. This choice is driven by the limited functionality of existing smart meters [NSM11],

and we believe that for a wide-scale deployment of NILM, existing infrastructure should be reused.

In addition, the research is not concern about the way in which the model parameters are obtained. This means that whether or not unsupervised learning is used is immaterial as long as the model parameters can be learned. The main emphasis is in the structure of the proposed model and its ability to address the limitations outlined in Section 2.7. However, we note that our proposed approach is agnostic to the learning paradigm. Should there be a need to obtain parameters in an unsupervised manner, existing approaches at tangent to our research emphases and which focus on the training aspect of NILM such as the work by Parson et al. [PGWR14] can be integrated. Though, such integration work is beyond the scope of this thesis.

Further, while the difficulty of detecting continuously-variable loads has been noted in the previous sections, and solutions have been limited thus far, these loads are not considered. Two assumptions are made in this regard. First, continuously-variable loads are uncommon in residential settings. Second, as described in an earlier section, Laughman et al. [LKC+03] specifically mentioned that their method could be used to remove continuously-variable components, enabling the remaining appliances to be detected in a conventional fashion. Therefore, in the event that such loads are present, it is assumed that future integration of the work by Laughman et al. [LKC+03] and the approach presented in this work would allow a more diverse set of appliances to be detected. Once again, the integration and the associated development are outside the scope of this thesis, as the main objective of this research lies in rectifying the issues stipulated in Section 2.7.

CHAPTER THREE

# MODELLING OF APPLIANCE BEHAVIOUR

In this chapter, an appliance model is formulated. With respect to the other established models used in the state-of-the-art, we define the main problem statement before outlining a set of requirements for the desired appliance model. It is with consideration of the stipulated requirements that the proposed model is devised. The outcome of this is the factorial variable transition hidden Markov model (FVTHMM), a factorial extension made for the model presented by [Vas91] and [RW92] in the field of speech modelling. Like the commonly-used explicit-duration hidden Markov model (EDHMM) [KAL11] [GWK15], FVTHMM is able to incorporate general state duration distributions. However, the differences in formulation inherent to FVTHMM allows a more natural integration with real-time and sequential processing requirements during the inference of appliances' states. This is achieved by describing the state sojourn time in terms of its hazard function. Finally, we present the results of modelling using the proposed model and how it could lead to better discernibility between appliances with similar power consumption. Part of this chapter has been published in a journal paper [WcD14].

## 3.1 Introduction

As noted in Chapter 2, disaggregation is a more challenging task when only aggregate real power data of low sampling rate measurements is available; the number of different electrical features that could be extracted to help distinguish between appliances is more limited than in the case of high sampling rate voltage and current waveform data. Therefore, better appliance models with less restrictive assumptions are required to compensate for this limitation.

For example, models characterising the behaviour of appliances should not only describe the relationship between the operational states of loads and the power consumption, but also incorporate correlation information which are apparent in the measurements at different times. While this may necessarily complicate the modelling process and increase the computational complexity of the subsequent disaggregation task, their inclusion is well-justified, given the need to improve separation between similar appliances and the lack of electrical features. Furthermore, the use of additional information in this way is appealing as no extra measurement device (e.g. light and temperature sensors etc.) is needed.

For this reason, the inclusion of temporal correlations has been popular amongst existing approaches that utilised only low sampling rate electrical measurements. A prime example is the use of state transition information in the form of hidden Markov models (HMM), of which instances have remained one of the better-performing methods in recent years. Yet, there is still room for improvements, considering the number of issues already outlined in Chapter 2. Particularly, the notion of states persisting over a variable period of time is not explicitly taken into account. Instead, this variation in state duration is implicitly restricted to be a geometric distribution, even though it may not be the case in reality. Such an example can be seen in Figure 3.1, where it is shown that the shape of the empirical probability density function associated with the ON state of a refrigerator is clearly different than that of the geometric distribution. The lack of expressive power in this regard limits its modelling capacity and prevents appliances with well-known cyclical usage patterns from being represented accurately. Additionally, with HMM, appliances with different state duration distribution could be modelled to have similar Markov state transition probabilities. This brings to light the issue of disambiguating between appliances that are alike with respect to first order statistics, despite the actual behavioural differences.

The solution to this, as has been explored in the literature, is the use of hidden semi-Markov models (HSMM). Durations of states, also known as sojourn time, are now allowed to take on arbitrary distributions while maintaining some aspects of Markov processes. As such, appliances with different characteristics could be represented more precisely to enable overlaps in power consumption features to be resolved. Following the same direction, we have adopted HSMM for NILM. However, unlike previous approaches which formulate the estimation as an off-line problem aiming to maximise the sequence likelihood that considers the probability density function of the observed sojourn times, the formulation presented herein allows dynamic computation of *time-varying* state transition

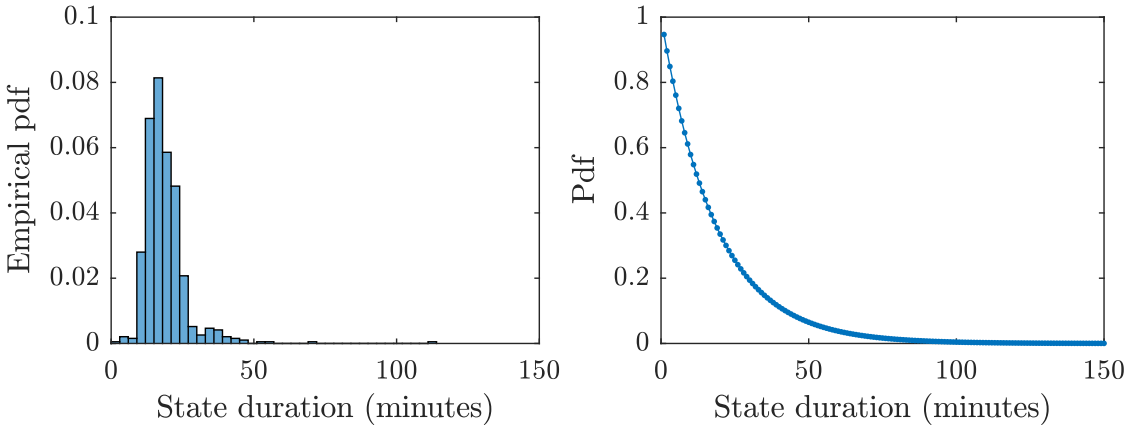



Figure 3.1: Differences between the actual duration distribution and the version implied through the use of HMM. Left: The empirical probability density function (pdf) associated with the ON state of a refrigerator. It has a mean ON duration of 19 minutes. Right: The corresponding geometric distribution with the same mean ON duration. Its shape clearly deviates from that of the actual duration distribution shown on the left.

probabilities at each time step conditioned on not just the previous state but also its dwell time, thereby facilitating on-line estimation of states. Moreover, state inference under the proposed model does not require explicit search over all possible state durations. This allows computation to be done efficiently, as we shall see in Chapter 4.

In this chapter, the following contributions are discussed:

- An alternative instance of HSMM for NILM with time-varying duration-dependent state transition probabilities.
- The use of a robust version of the Expectation-Maximisation (EM) algorithm for learning the associations between the states of appliances and their power consumption in the presence of outlying values originating from transients. It is often not mentioned in the literature how these values are being dealt with.
- The automatic determination of the number of mixtures in the state duration distribution using an information criterion known as the Minimum Message Length (MML) principle.

## 3.2 Related Work

The existing use of state duration for load disaggregation is limited. The main work is that of Kim et al. [KAL11] and of Johnson and Willsky [JW13], which have

both demonstrated that the explicit modelling of state duration using HSMM does indeed help to improve disaggregation accuracy.

Suppose that the state of appliance $k$ at time $t$ is $x_{t,k}$ and suppose that $\mathbf{x}_t = (x_{1,1}, \ldots, x_{t,K})$ denotes the system state composed of $K$ appliances. Given a sequence of $T$ system states $\mathbf{x}_{1:T}$ and a sequence of $T$ aggregate measurements $y_{1:T}$, the joint probability characterising the important parts of the model described in these approaches is

$$p(\mathbf{x}_{1:T}, y_{1:T}) = \prod_{t=1}^{T} p(y_t \mid \mathbf{x}_t) \prod_{k=1}^{K} \Bigg[ p(x_{1,k}) p(d_{1,k} \mid x_{1,k}) \times \prod_{r: x_{r,k} \neq x_{r-1,k}} p(x_{r,k} \mid x_{r-1,k}) p(d_{r,k} \mid x_{r,k}) \Bigg], \tag{3.1}$$

where $p(d_{r,k} \mid x_{r,k})$ is the probability that, for $d_{r,k}$ consecutive time steps from $r$, the state remains unchanged, i.e. $x_{\tau,k} = x_{r,k}$ for $\tau = r+1, \ldots, d_{r,k} - 1$ and $x_{r,k} \neq x_{r+d_{r,k},k}$.

Intuitively, (3.1) can be understood as a process in which $\mathbf{x}_{1:T}$ and $y_{1:T}$ are generated. At the start, and for each $k$, $x_{1,k}$ is drawn from the distribution $p(x_{1,k})$. Then, subject to the realisation of $x_{1,k}$, $d_{1,k}$ is sampled from $p(d_{1,k} \mid x_{1,k})$ so that $x_{\tau,k} = x_{1,k}$ for $\tau = 2, \ldots, d_{1,k} - 1$. After these $d_{1,k}$ states are generated, a new state of appliance $k$ at time $r$ is drawn from $p(x_{r,t} \mid x_{r,t-1})$, with the constraint that a different state than before has to be chosen, i.e. $x_{r,k} \neq x_{r,k-1}$. Subsequently, like before, the duration corresponding to $x_{r,k}$ is drawn from $p(d_{r,k} \mid x_{r,k})$, resulting in $x_{\tau,k} = x_{r,k}$ for $\tau = r+1, \ldots, r + d_{r,k} - 1$. In the end, having generated a sequence of $T$ states for each $k$, $T$ observations are each sampled independently according to $p(y_t \mid \mathbf{x}_t)$.

The formulation of the aforementioned generative process is a specific instance of HSMM, known as the explicit duration hidden Markov model (EDHMM) [Yu10]. While its description is relatively simple to understand, the direct usage of the state duration distribution $p(d_{r,k} \mid x_{r,k})$ in (3.1) does not easily allow real-time inference of states. For example, the maximisation of (3.1) with respect to $\mathbf{x}_{1:T}$ using the Viterbi algorithm not only has to be done over the set of possible states at each time step, but also over a potentially large number of durations. This is on top of the typically required hard-coded bounds on the possible durations to be explored so that memory cost is bounded. Unless methods borrowed from the field of Bayesian non-parametrics are used [DWW12], setting bounds in advance could result in wrong inferences, besides being not ro-

bust against unforeseen situations. Overall, these drawbacks prevent the use of EDHMM in meeting the research objectives stipulated in Chapter 1.

To that end, an alternative formulation of HSMM with time-varying duration-dependent state transition probabilities is proposed for NILM. Details pertaining to it are described in the next section.

## 3.3 Time-Varying State Transition Probabilities

Independently developed as part of this research and later found to have been used in the field of acoustic speech modelling [Vas91, RW92], the alternative formulation of HSMM, also known as the variable transition hidden Markov model (VTHMM) [Yu10], sets itself apart from EDHMM with the inclusion of duration-dependent state transition probabilities, i.e. $p(x_{t,k} \mid x_{t-1,k}, c_{t-1,k})$ where $c_{t-1,k}$ denotes the number of time steps spent in state $x_{t-1,k}$. In other words, the state transition probability is no longer static as is the case with HMM and EDHMM. Instead, it is now a function of $c_{t-1,k}$. Although it was noted by Johnson [Joh05] that both EDHMM and VTHMM are equivalent in modelling a given $y_{1:T}$, the process of performing state inference under both models is not exactly the same [Yu10].

### 3.3.1 Factorial Variable Transition HMM

Here, we propose a factorial variant of VTHMM, factorial variable transition HMM (FVTHMM[1]), for load disaggregation. With $K$ independent chains (e.g. $K$ appliances), the model is characterised by the joint probability

$$\begin{aligned} p(\mathbf{x}_{1:T}, y_{1:T}, \mathbf{c}_{1:T}) = p(\mathbf{x}_1)p(\mathbf{c}_1) \prod_{t=1}^{T} p(y_t \mid \mathbf{x}_t) \\ \times \prod_{r=2}^{T} p(\mathbf{x}_r \mid \mathbf{x}_{r-1}, \mathbf{c}_{r-1}) p(\mathbf{c}_t \mid \mathbf{x}_r, \mathbf{c}_{r-1}, \mathbf{x}_{r-1}) \end{aligned} \tag{3.2}$$

where $\mathbf{x}_t \in \mathbb{Z}^{1\times K}$ refers to the system state at time $t$ as before, i.e. $\mathbf{x}_t = (x_{t,1}, \ldots, x_{t,K})$; $y_t \in \mathbb{R}$ is the aggregate real power as measured at time $t$ and $\mathbf{c}_t \in \mathbb{Z}^{1\times K}$ is a vector of counters corresponding to the $K$ appliances, i.e. $\mathbf{c}_t = (c_{t,1}, \ldots, c_{t,K})$. Like any factorial model, due to the independence relationship between chains, the factors $p(\mathbf{x}_1)$, $p(\mathbf{c}_1)$, $p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{c}_{t-1})$ and $p(\mathbf{c}_t \mid \mathbf{x}_t, \mathbf{c}_{t-1}, \mathbf{x}_{t-1})$ are

[1] To ease readability, FVTHMM can be pronounced as "fathom".

$$p(\mathbf{x}_1) = \prod_{k=1}^{K} p(x_{1,k}) \tag{3.3}$$

$$p(\mathbf{c}_1) = \prod_{k=1}^{K} p(c_{1,k}) \tag{3.4}$$

$$p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{c}_{t-1}) = \prod_{k=1}^{K} p(x_{t,k} \mid x_{t-1,k}, c_{t-1,k}) \tag{3.5}$$

$$p(\mathbf{c}_t \mid \mathbf{x}_t, \mathbf{c}_{t-1}, \mathbf{x}_{t-1}) = \prod_{k=1}^{K} p(c_{t,k} \mid x_{t,k}, c_{t-1,k}, x_{t-1,k}). \tag{3.6}$$

respectively. The joint probability in (3.2) can also be expressed in a recursive form,

$$\begin{aligned} p(\mathbf{x}_{1:t}, y_{1:t}, \mathbf{c}_{1:t}) = {} & p(\mathbf{x}_{1:t-1}, y_{1:t-1}, \mathbf{c}_{1:t-1}) p(y_t \mid \mathbf{x}_t) \\ & \times p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{c}_{t-1}) p(\mathbf{c}_t \mid \mathbf{x}_t, \mathbf{c}_{t-1}, \mathbf{x}_{t-1}), \end{aligned} \tag{3.7}$$

to enable the incremental computation of probabilities. This will be used in the proposed state inference method in Chapter 4.

To understand how the FVTHMM works as a generative model, consider its dynamic Bayesian network (DBN) representation as shown in Figure 3.2. For each of the $K$ chains, and for each $t$, the realisation of $x_{t,k}$ is conditionally dependent on the previous state $x_{t-1,k}$ and its dwell time $c_{t-1,k}$. This relationship is

$$p(x_{t,k} \mid x_{t-1,k}, c_{t-1,k}) = \begin{cases} h_{x_{t-1,k}}(c_{t-1,k}) \tilde{a}_{x_{t-1,k}, x_{t,k}}, & \text{if } x_{t,k} \neq x_{t-1,k} \\ 1 - h_{x_{t-1,k}}(c_{t-1,k}), & \text{otherwise,} \end{cases} \tag{3.8}$$

where

- $h_{x_{t-1,k}}(c_{t-1,k})$ is the hazard function from the field of survival analysis [Jen05], which equivalently refers to the probability of exiting the state $x_{t-1,k}$, having already spent $c_{t-1,k}$ time steps in that state, i.e.

$$h_{x_{t-1,k}}(c_{t-1,k}) = \frac{p(d = c_{t-1,k} \mid x_{t-1,k})}{p(d \geq c_{t-1,k} \mid x_{t-1,k})}; \tag{3.9}$$

- $p(d \mid x_{t-1,k})$ is the duration probability of the state $x_{t-1,k}$;

- for $x_{t-1,k} = i$ and $x_{t,k} = j$, $\tilde{a}_{i,j}$ is the probability of transitioning from state $i$ to state $j$ conditioned on $i$ being different than $j$. As such, if appliance $k$ has $M_k$ states, $\tilde{a}_{i,j}$ can be seen as an element in the $i$th row and $j$th column of

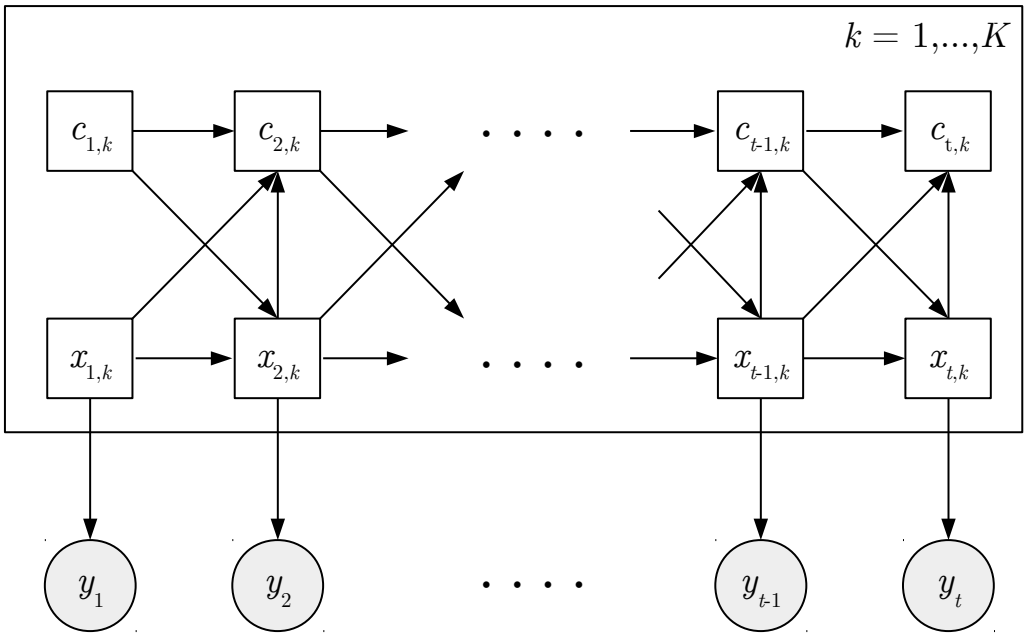



Figure 3.2: Dynamic Bayesian network of the FVTHMM.

a state transition matrix $\tilde{A}_k \in [0,1]^{M_k \times M_k}$ such that $\sum_j \tilde{a}_{i,j} = 1$ and $\tilde{a}_{i,i} = 0$. Note that $\tilde{A}_k$ can be derived from the state transition matrix $A_k \in [0,1]^{M_k \times M_k}$ of an ordinary discrete time HMM, as for any $i \neq j$,

$$\tilde{a}_{i,j} = \frac{a_{i,j}}{\sum_{j \neq i} a_{i,j}}, \tag{3.10}$$

where $a_{i,j}$ is the corresponding element in matrix $A_k$.

The counter $c_{t,k}$, on the other hand, takes on the value of $c_{t-1,k} + 1$ if $x_{t,k} = x_{t-1,k}$. Otherwise, it resets to 1. This means, the probability $p(c_{t,k} \mid x_{t,k}, c_{t-1,k}, x_{t-1,k})$ is a sparse distribution since other counter values are not possible by construction. Therefore, the conditional dependence of $c_{t,k}$ on $x_{t,k}$, $c_{t-1,k}$ and $x_{t-1,k}$ can be formally presented as

$$p(c_{t,k} \mid x_{t,k}, c_{t-1,k}, x_{t-1,k}) = \begin{cases} \delta_{c_{t,k}, c_{t-1,k}+1}, & \text{if } x_{t-1,k} = x_{t,k} \\ \delta_{c_{t,k},1}, & \text{otherwise,} \end{cases} \tag{3.11}$$

where $\delta$ denotes the Kronecker delta function,

$$\delta_{i,j} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{otherwise.} \end{cases} \tag{3.12}$$

We assume that the counter starts at 1 when the initial state is entered, i.e. $p(c_{1,k} = 1) = 1$ for all $k$.

The relationship between the aggregate power consumption at time $t$, $y_t$, and the system state $\mathbf{x}_t$ is same as that of the factorial EDHMM (FEDHMM) and FHMM, as the observed measurement at the mains is clearly dependent on the internal state of each of the $K$ appliances in a residential unit, i.e. $p(y_t \mid \mathbf{x}_t) = p(y_t \mid x_{t,1}, \ldots, x_{t,K})$.

### 3.3.2 FVTHMM as Applied to NILM

#### Modelling of Power Consumption

In NILM, the fundamental equation relating the aggregate power consumption, $y_t$, to the individual appliance power consumption, $y_{t,k}$, is

$$y_t = \sum_{k=1}^{K} y_{t,k} + r_t, \tag{3.13}$$

where $r_t$ is a general noise term that might include noise induced by measurement errors or contributions from appliances for which we are unaware. Though, for now, we will assume that $r_t$ is 0, implying we have knowledge of all appliances in a residential unit. Cases, where some appliances are unknown, are addressed in Chapter 5.

On its own, the power consumption of appliance $k$ at time $t$, $y_{t,k}$, is naturally a function of its operational state, $x_{t,k}$, i.e. $f_k(x_{t,k})$. It also has a noise term that is dependent on $x_{t,k}$, i.e. $n_k(x_{t,k})$. For example, a three-state fan might have larger fluctuations about its nominal power consumption when it is operating in the highest speed, while the fluctuations might be lower when a medium speed is chosen. Taken together, the power consumption of appliance $k$ can be expressed as

$$y_{t,k} = f_k(x_{t,k}) + n_k(x_{t,k}). \tag{3.14}$$

The value of $f_k(x_{t,k})$ is constant for a given $x_{t,k}$ and it is deterministic, whereas the state-dependent noise of appliance $k$, $n_k(x_{t,k})$, is a random variable.

It is usually not the case that $n_k(x_{t,k})$ is independent and identically distributed (i.i.d.) for a fixed $x_{t,k}$. Refrigerators, for instance, usually have a decaying power consumption from the moment they enter their ON cycle (see Figure 3.3). Therefore, for a given $x_{t,k}$, it may be more appropriate to model their power consumption as a non-stationary process. Though, for the purpose of this chapter, we

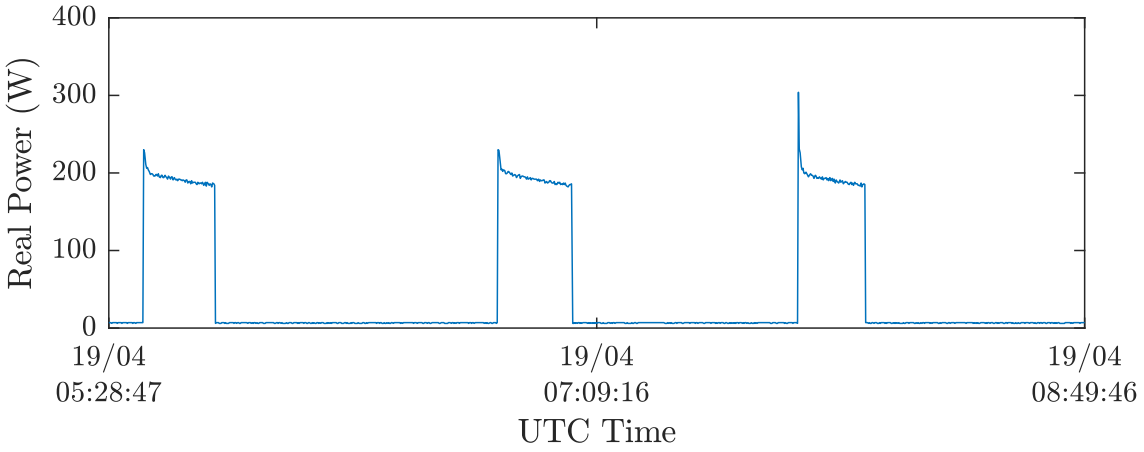



Figure 3.3: Multiple ON cycles of a refrigerator. The decay in power consumption from the onset of each cycle is apparent. The data is from house 1 of the REDD dataset.

will follow previous work [KAL11, KJ12] and assume that the state-dependent noise for each appliance is i.i.d.. Deviation from this assumption is explored in Section 4.5 of Chapter 4. Additionally, $n_k(x_{t,k})$ is also assumed to have a Gaussian distribution, like in existing approaches [KAL11, KJ12, PGWR14]. This means,

$$y_{t,k} \mid x_{t,k} \sim \mathcal{N}(\mu_{x_{t,k}}, \sigma^2_{x_{t,k}}), \tag{3.15}$$

where $f_k(x_{t,k}) = \mu_{x_{t,k}}$ is the mean of power consumption associated with state $x_{t,k}$ of appliance $k$ and $\sigma^2_{x_{t,k}}$ is the corresponding variance.

However, because $y_{t,k}$ cannot be observed in NILM, we are only interested in the relationship between the aggregate power consumption $y_t$ and the internal state of each appliance. For that, and using the fact that the summation of $K$ independent Gaussian distributed random variables is also a Gaussian random variable, we arrive at

$$y_t \mid \mathbf{x}_t \sim \mathcal{N}\left(\sum_{k=1}^{K} \mu_{x_{t,k}}, \sum_{k=1}^{K} \sigma^2_{x_{t,k}}\right), \tag{3.16}$$

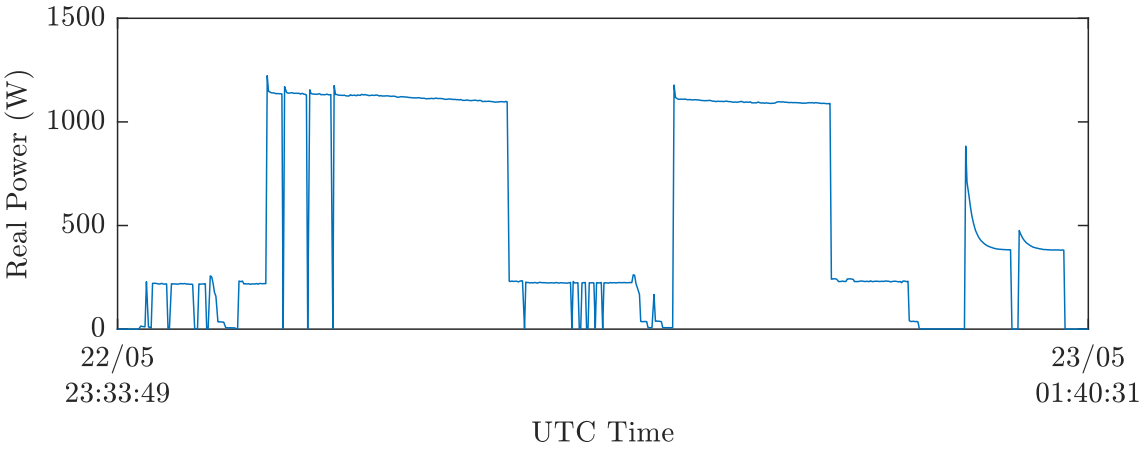
which is used as the emission probability, $p(y_t \mid \mathbf{x}_t)$, of FVTHMM in (3.2).

### Modelling of Appliance State Sojourn Time

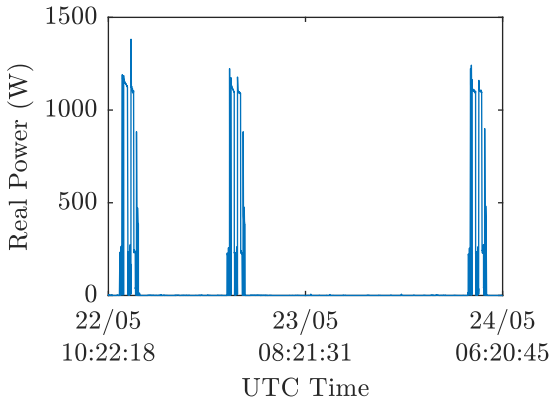
The state duration probability, as used in the hazard function calculation in (3.9), is represented using a mixture of Gamma distributions. The choice of using a mixture instead of a single Gamma distribution is driven by the observation that some appliances are associated with both long periods without use and short-

term OFF periods that are embedded within an operating cycle. For example, Figure 3.4a shows that there are multiple short OFF periods within an operating cycle of a dishwasher, while Figure 3.4b illustrates the longer inter-usage period between operating cycles. Overall, this operational behaviour is apparent if we consider the duration distribution of the OFF state, as shown in Figure 3.4c; the large peak at around 0 is attributed to the short OFF periods and the cluster centring at 1 suggests that the dishwasher is normally being operated once every day.

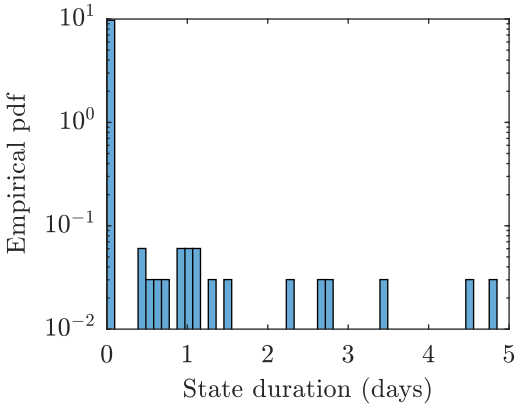
As a distinction is needed between the two (or possibly more) scenarios, a mixture of distributions is appropriate; one mixture component is for modelling the long-term OFF periods while another is for representing the short-term OFF periods. Thus, for the duration $d$ of a given state $x_{t,k} = i$ of a certain appliance $k$,




(a) Power consumption of a dishwasher in one operating cycle.




(b) Zoom-out version of Figure 3.4a showing the actual inter-usage period.




(c) Duration distribution of the dishwasher's OFF state

Figure 3.4: Operational behaviour of a dishwasher. The power consumption data is from house 1 of the REDD dataset [KJ11].

a mixture of $L_i$ Gamma distributions is

$$p(d \mid x_{t,k} = i) = \sum_{l=1}^{L_i} m_{l,i} g(d; \alpha_{l,i}, \beta_{l,i}), \tag{3.17}$$

where

$$g(d; \alpha_{l,i}, \beta_{l,i}) = \frac{d^{\alpha_{l,i}} \exp(-d/\beta_{l,i})}{\beta_{l,i}^{\alpha_{l,i}} \Gamma(\alpha_{l,i})} \tag{3.18}$$

is the probability density function (pdf) of the $l$th component Gamma distribution, parametrised by the shape parameter $\alpha_{l,i}$ and the scale parameter $\beta_{l,i}$; $m_{l,i}$ is the mixing coefficient of the $l$th component, with the constraint that $\sum_{l=1}^{L_i} m_{l,i} = 1$ and $\forall l, 0 \leq m_{l,i} \leq 1$; and $\Gamma(\cdot)$ is the Gamma function.

## 3.4 Learning of Model Parameters

Before describing the approach used for learning model parameters, let us first define some notations to simplify the discussion that follows. Firstly, for appliance $k$ with a set of disjoint states $\mathcal{M}_k$ whose cardinality is $M_k = |\mathcal{M}_k|$, we shall denote the mean and variance of the power consumption of state $i$ by $\mu_i$ and $\sigma_i^2$. Accordingly, a collection of such means and variances for all $M_k$ states of appliance $k$ is referred to as $(\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k^2) = [(\mu_i, \sigma_i^2)]_{i \in \mathcal{M}_k}$. Secondly, for the state transition model, the Markov state transition matrix associated with appliance $k$ will be designated as $A_k$, while $\tilde{A}_k$ is the corresponding state transition matrix with zeroed self-transition probabilities. Also relevant is the initial probability of being in state $i$, $\pi_i = p(x_{1,k} = i)$, with $\boldsymbol{\pi}_k = [\pi_i]_{i \in \mathcal{M}_k}$. Thirdly, a mixture of $L_i$ Gamma distributions for modelling the durations of state $i$ of appliance $k$ has the parameters $\Theta_i = [(m_{l,i}, \alpha_{l,i}, \beta_{l,i})]_{l=1}^{L_i}$, where $m_{l,i}$, $\alpha_{l,i}$ and $\beta_{l,i}$ are the mixture coefficient, the shape parameter and scale parameter of the $l$th component respectively. For appliance $k$, we then have $(\boldsymbol{m}_k, \boldsymbol{\alpha}_k, \boldsymbol{\beta}_k) = [\Theta_i]_{i \in \mathcal{M}_k}$. Taken together, the complete model parameters of a $K$-chain factorial VTHMM can be represented by the tuple $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_{\mathrm{e}}, \boldsymbol{\lambda}_{\mathrm{d}})$, with $\boldsymbol{\lambda}_{\mathrm{e}}$ being the tuple comprising of the parameters related to the emission probability, and $\boldsymbol{\lambda}_{\mathrm{d}}$ being that of the state transition probability and the state transition model, i.e. $\boldsymbol{\lambda}_{\mathrm{e}} = [(\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k^2)]_{k=1}^K$ and $\boldsymbol{\lambda}_{\mathrm{d}} = [(\boldsymbol{\pi}_k, A_k, \boldsymbol{m}_k, \boldsymbol{\alpha}_k, \boldsymbol{\beta}_k)]_{k=1}^K$.

These parameters can be learned using the Expectation-Maximisation (EM) algorithm briefly described in Section 2.5.3, when only a sequence of $T$ aggregate measurements $y_{1:T}$ is available and both the corresponding system states $\mathbf{x}_{1:T}$ and the appliance-level contributions $y_{1:T,k}$ are unknown. However, this constitutes an fully-unsupervised learning problem and it is beyond the scope of this

research. Instead, we will consider the case where training data in the form of $y_{1:T,k}$ is available but the states of appliance $k$, $x_{1:T,k}$, are unknown. While this is a less elegant approach, we again note that the main emphasis of this thesis is in the efficient and robust inference of states under a more complex model with duration modelling. To that end, the formulation of the EM algorithm for fully unsupervised learning under FVTHMM is reserved for future work.

In this section, a supervised learning approach used for finding $\boldsymbol{\lambda}$ is described, whereby we first detail the technique used for estimating $\boldsymbol{\lambda}_{\mathrm{e}}$ in the presence of outlying values caused by transients, before discussing the method of inferring $\boldsymbol{\lambda}_{\mathrm{d}}$. Henceforth, we shall also refer to $\boldsymbol{\lambda}_{\mathrm{e}}$ and $\boldsymbol{\lambda}_{\mathrm{d}}$ as the parameters for the emission model and the temporal model respectively, given that the former is concerned with the emission of an observed value for a given state while the latter governs the dynamics of the states with respect to time.

### 3.4.1 Parameter Estimation for the Emission Model

For the purpose of estimating $\boldsymbol{\lambda}_{\mathrm{e}}$, the power consumption of appliance $k$ is fitted using a mixture of $t$-distributions, with each mixture component corresponding to a state. This choice was made, as it was found that the learning of parameters based on a mixture of Gaussian distributions is sensitive to outlying values of power consumption, which originate often from transients. Moreover, it has been mentioned by Lucas [Luc97] and Peel and McLachlan [PM00] that the use of $t$-distributions is closely related to employing a well-known technique called the "M-estimators" for robust estimation of parameters. Therefore, the approach considered here first estimates the parameters of a mixture of $t$-distributions, before converting these inferred parameters into their Gaussian equivalent.

We note that the conversion is not necessarily required and the $t$-distributions can be used as it is. However, unlike the simple expression in (3.16), there are complications in deriving the distribution of the sum of $t$-distributed random variables from the parameters of the distribution of the summands; the sum is no longer a $t$-distribution. The only exception is the sum of random variables, each distributed according to a $t$-distribution with degree-of-freedom 1 (i.e. a Cauchy distribution), but such considerations are reserved for future work, given the time constraint of this research.

For a sequence of $T$ power measurements of appliance $k$, $y_{1:T,k}$, from a training dataset, the probability density function of a mixture of $t$-distributions with $M_k$

components is given as

$$p(y_{t,k} \mid [(\omega_i, \mu_i, \kappa_i^2, \nu_i)]_{i=1}^{M_k}) = \sum_{i=1}^{M_k} \omega_i \phi(y_t; \mu_i, \kappa_i^2, \nu_i), \tag{3.19}$$

where

$$\phi(y_t; \mu_i, \kappa_i^2, \nu_i) = \frac{\Gamma\left(\frac{\nu_i+1}{2}\right)}{(\pi\nu_i)^{1/2}\Gamma(\nu_i/2)\kappa_i \left[1 + \frac{1}{\nu_i}\left(\frac{y_t-\mu_i}{\kappa_i}\right)^2\right]^{\frac{(\nu_i+1)}{2}}} \tag{3.20}$$

is the probability density function of the $i$th component $t$-distribution, parametrised by the corresponding location parameter $\mu_i$, scale parameter $\kappa_i$ and the degree-of-freedom parameter $\nu_i$; $\omega_i$ refers to the mixing coefficient of the $i$th component. The degree-of-freedom parameter can be interpreted as a tuning parameter for controlling the degree of robustness [Luc97]; a small $\nu_i$ implies higher robustness or vice versa. In modelling the power consumption of appliances, we have chosen to fix $\nu_i$ at 3 for all $i$. Thus, the parameters to be estimated from $T$ power measurements of appliance $k$ are $\boldsymbol{\theta}_{\mathrm{e},k} = \{(\omega_i, \mu_i, \kappa_i^2)\}_{i=1}^{M_k}$, with $\nu_i$ omitted. Note that the number of states for appliance $k$, $M_k$, is determined through the number of peaks in the histogram. However, other methods like the one described in [KDM+16] could be used.

To meet the goal of inferring these parameters, a robust version of EM algorithm for a mixture of $t$-distributions is adopted from the work of Peel and McLachlan [PM00]. Before describing the method, we should emphasise that the EM algorithm as used here differs from the EM algorithm for FVTHMM mentioned at the start of Section 3.4, as the latter jointly estimates both $\boldsymbol{\lambda}_{\mathrm{e}}$ and $\boldsymbol{\lambda}_{\mathrm{d}}$ from only the aggregate measurements, while the former infers parameters of the emission model from the appliance-level contributions of a training dataset. Only the important aspects of the robust EM algorithm to find $\boldsymbol{\theta}_{\mathrm{e},k}$ are presented here. For a more detailed exposition and a complete derivation of the relevant expressions, see [PM00].

In any case, given an initial guess of $\boldsymbol{\theta}_{\mathrm{e},k}$ or an estimate from the previous iteration $n$, $\boldsymbol{\theta}_{\mathrm{e},k}^{[n]}$, the E-step of the robust EM algorithm involves calculating the soft assignment probabilities of each power consumption measurement $y_{t,k}$ to the components $i = 1, \ldots, M_k$,

$$\tau_{it}^{[n]} = \frac{\omega_i^{[n]} \phi_i(y_t; \mu_i^{[n]}, \kappa_i^{[n]}, \nu_i = 3)}{\sum_{i=1}^{M_k} \omega_i^{[n]} \phi_i(y_t; \mu_i^{[n]}, \kappa_i^{[n]}, \nu_i = 3)} \tag{3.21}$$

and the weights

$$u_{it}^{[n]} = \frac{\nu_i + 1}{\nu_i + \left(\frac{y_t - \mu_i^{[n]}}{\kappa_i^{[n]}}\right)^2}. \tag{3.22}$$

Then, in the M-step, the estimates are updated using

$$\omega_i^{[n+1]} = \frac{\sum_{t=1}^{T} \tau_{it}^{[n]}}{T} \tag{3.23}$$

$$\mu_i^{[n+1]} = \frac{\sum_{t=1}^{T} \tau_{it}^{(k)} u_{it}^{[n]} y_t}{\sum_{t=1}^{T} \tau_{it}^{[n]} u_{it}^{[n]}} \tag{3.24}$$

$$(\kappa_i^2)^{[n+1]} = \frac{\sum_{t=1}^{T} \tau_{it}^{(k)} u_{it}^{[n]} (y_t - \mu_i^{[n+1]})^2}{\sum_{t=1}^{T} \tau_{it}^{[n]}}. \tag{3.25}$$

Multiple iterations of such computations are performed until the likelihood $\prod_{t=1}^{T} p(y_t \mid [(\omega_i, \mu_i, \kappa_i^2, \nu_i)]_{i=1}^{M_k})$ converges, upon which the last computed parameters are taken to be the final estimates, $\hat{\boldsymbol{\theta}}_{\mathrm{e},k} = [(\hat{\omega}_i, \hat{\mu}_i, \hat{\kappa}_i^2)]_{i=1}^{M_k}$.

Finally, the Gaussian parameters in $(\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k^2)$ are derived from $\hat{\boldsymbol{\theta}}_{\mathrm{e}}$ such that the mean of state $i$ of appliance $k$ is the estimated location parameter of the $i$th component, $\hat{\mu}_i$, and the variance of state $i$ is the variance of the $i$th component $t$-distribution, i.e. $\sigma_i^2 = \nu_i(\nu_i - 2)^{-1}\hat{\kappa}_i^2$. The estimated mixing coefficients $[\hat{\omega}_i]_{i=1}^{M_k}$ are not used.

The reduced sensitivity to outliers can be understood from the role of $\nu_i$ and $u_{it}$ in influencing the estimates of $\mu_i$ and $\kappa_i^2$. For small values of $\nu_i$, the Mahalanobis squared distance, $(\frac{y_t - \mu_i}{\kappa_i})^2$, in the denominator of (3.22) greatly reduces the value of $u_{it}$ if a certain data point $y_t$ is far away from the location parameter $\mu_i$. Therefore, as can be seen in (3.24) and (3.25), the influence of such data points are downweighted when computing for the new iteration's $\mu_i$ and $\kappa_i^2$. Whereas, if $\nu_i$ is large, the values of $u_{it}$ will be less affected by the Mahalanobis squared distance; in the limit of $\nu_i \to \infty$, $u_{it}$ becomes 1, leading to the case where all data points are equally weighted and thus, falling back to an ordinary non-robust EM algorithm for a mixture of Gaussian distributions. This is how the value of $\nu_i$ controls the robustness of the estimates to potential outlying values in the training dataset.

### Examples of application to real-world appliance data

The direct application of an ordinary EM algorithm for a mixture of Gaussian distributions (henceforth referred to the non-robust EM) is not robust to anomalies which potentially exist in real-world data. Figure 3.5 shows an example of one such anomaly from the REDD dataset, where the highlighted value is not actually part of the nominal power consumption but an intermediate value that was sampled by chance while the appliance transitioned from the ON state to the OFF state.

Even though the occurrence of these outlying observations is not as common as that of the normal measurements for this particular appliance (see Figure 3.6), they are able to influence the fitted distribution obtained via the non-robust EM greatly. Specifically, the second mixture component of the distribution has a slightly left-shifted mean and a large variance that is not reflective of

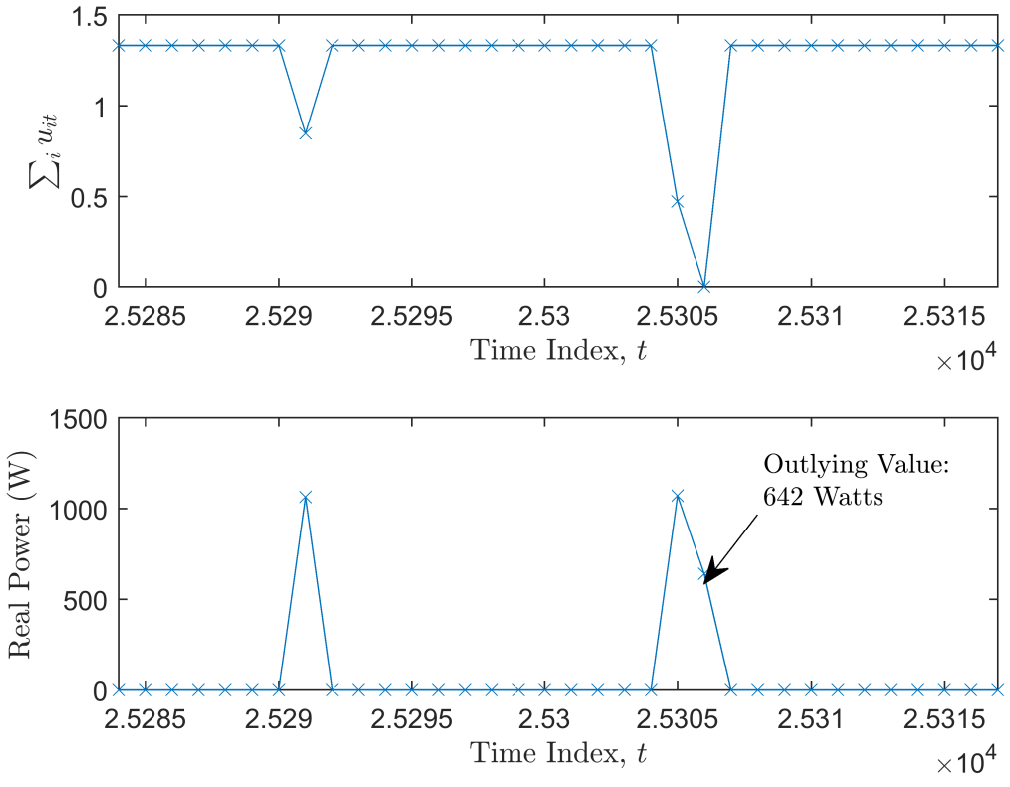



Figure 3.5: An anomalous measurement. The operating power of 642 Watts for the second kitchen outlet of house 2 of the REDD dataset has a $\sum_i u_{it}$ value of approximately zero, indicating that this is potentially an outlier. In fact, upon closer inspection, this operating power is the intermediate value sampled during the ON-to-OFF transition. Thus, as far as steady-state operation is concerned, it can be considered an anomaly.

the actual operation of the appliance. This is especially apparent, if we consider the quantile-quantile plots for the robustly fitted distribution and the non-robust counterpart shown in Figure 3.7a and Figure 3.7b. Ideally, the points should lie on the diagonal line. However, as shown in Figure 3.7b, there is a large deviation; a large number of data points fall into the quantiles corresponding to the second mixture component, indicating that the non-robustly fitted distribution deviates significantly from the underlying distribution inherent to the data.

Similar results can also be seen from fitting a distribution over the data of a refrigerator, as shown in Figure 3.8. Particularly, the intermediate power values between state transitions appear to smear two mixture components of the non-robustly fitted distribution, shifting away their means from the two clusters of data centring at about 350W and 420W, before merging into one partially overlapping component with mass spreading across a wide range of values. Additionally, from the quantile-quantile plot of the non-robust EM in Figure 3.9b, the large number of points deviating from the $45^\circ$ line suggests that the distribution is forced to stretch across horizontally to accommodate the outlying observations, at the expense of modelling each cluster accurately.

On the other hand, the few points associated with the upper tail-end of the data, as illustrated in Figure 3.9a and Figure 3.9b, do not seem to fit well for both that of the robust EM and the non-robust EM as the transients at the onset

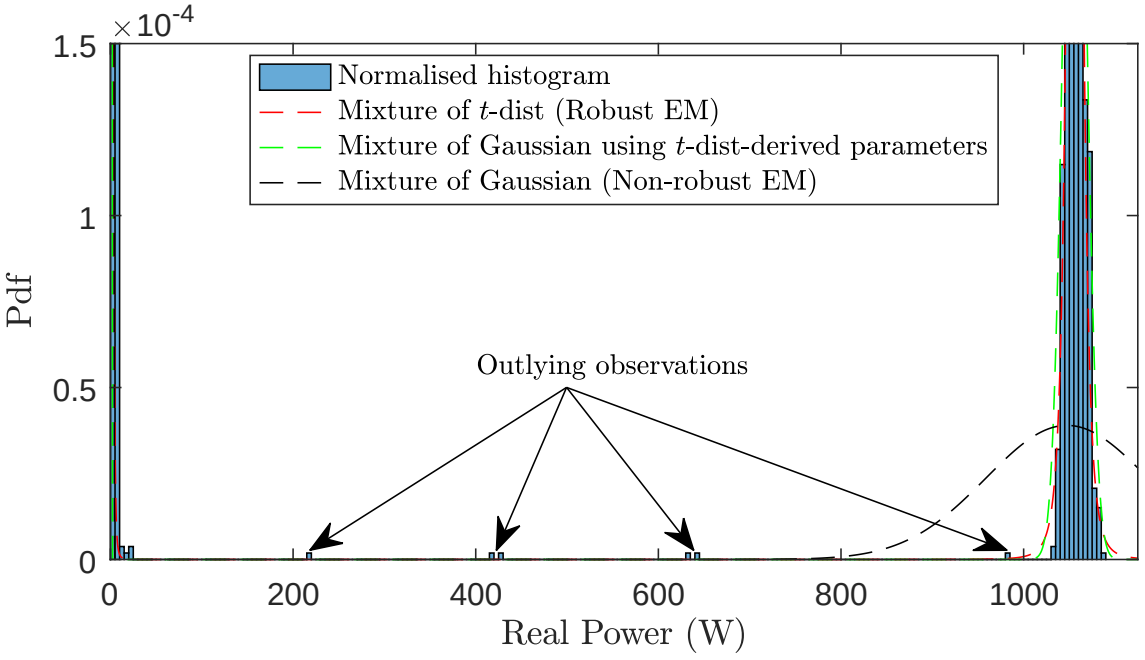
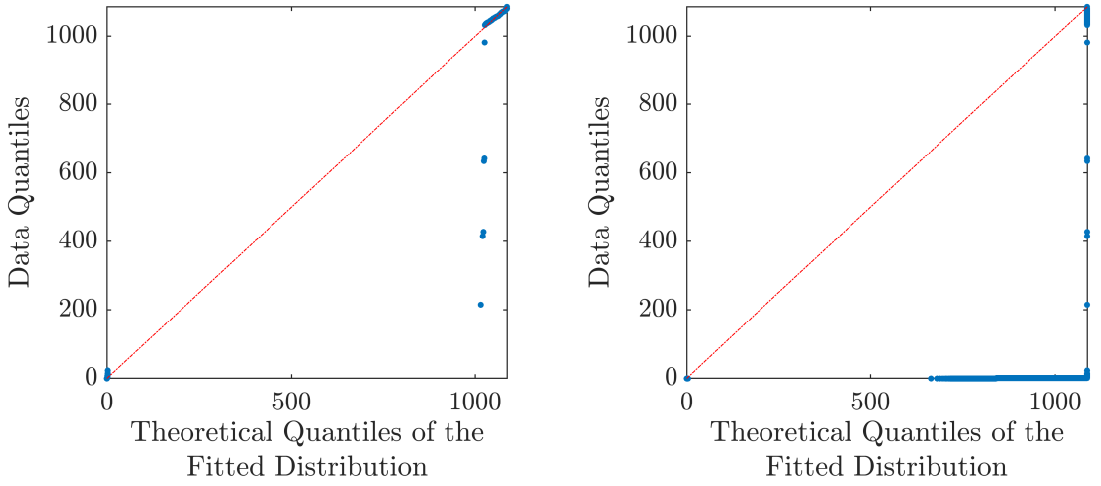



Figure 3.6: Distribution-fitting on the second kitchen outlet of house 2 of the REDD dataset using robust EM (mixture of $t$-dist) vs non-robust EM (mixture of Gaussian).




(a) Using the robust EM algorithm

(b) Using the non-robust EM algorithm

Figure 3.7: Quantile-quantile plot of the second kitchen outlet of house 2 of the REDD dataset with the fitted distribution.

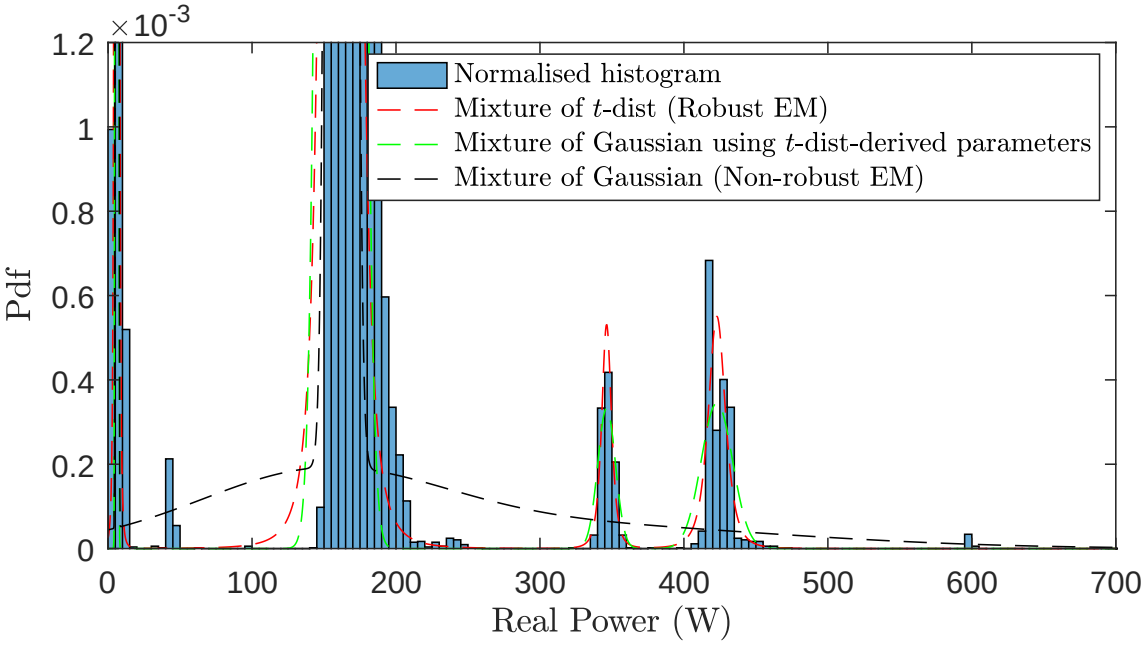
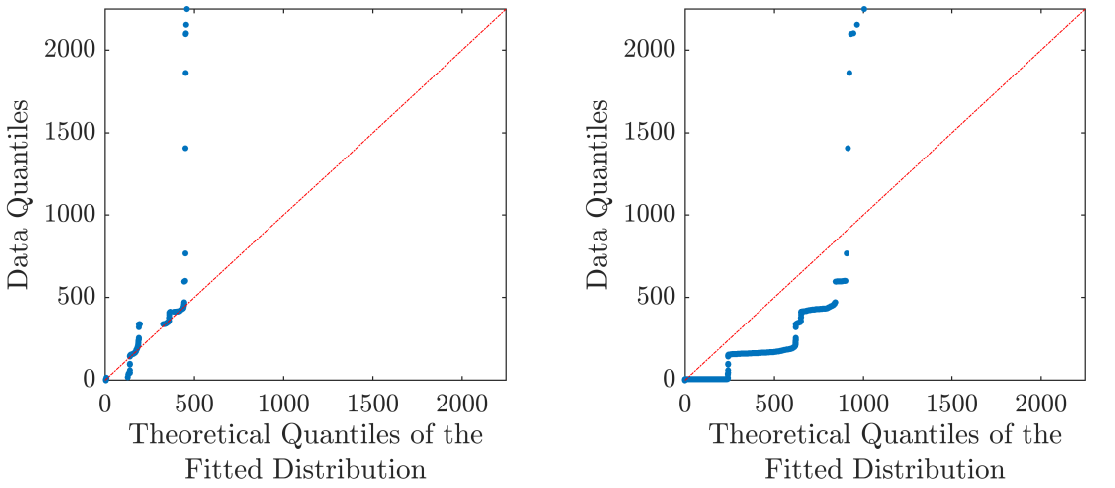



Figure 3.8: Distribution-fitting on the refrigerator of house 2 of the REDD dataset using robust EM (mixture of $t$-dist) vs non-robust EM (mixture of Gaussian).




(a) Using the robust EM algorithm (b) Using the non-robust EM algorithm

Figure 3.9: Quantile-quantile plot of the refrigerator of house 2 of the REDD dataset with the fitted distribution.

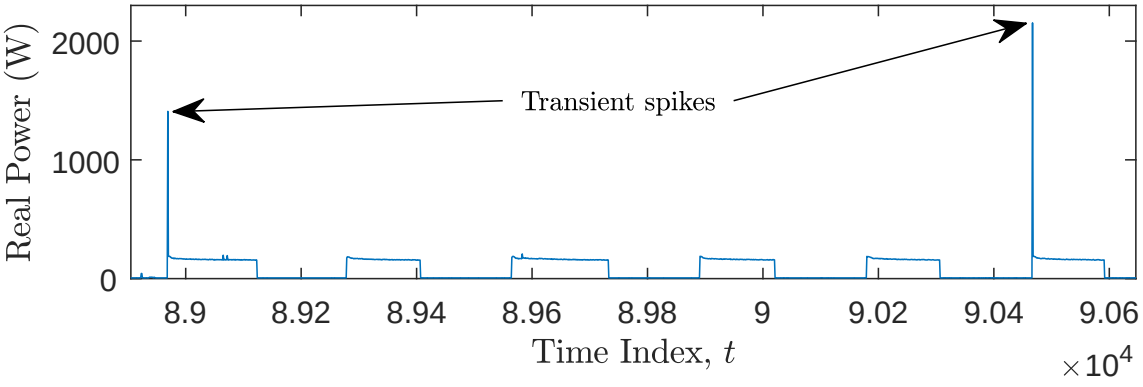



Figure 3.10: The large variation in transient spikes of the refrigerator.

of each operating cycle of the refrigerator vary over a large range of values (see Figure 3.10). However, the robustly fitted distribution over the other clusters is clearly less affected by these anomalies, judging from the relatively large number of data points close to the $45^\circ$ line.

One other interesting observation of the refrigerator data is the skewness of distribution of the cluster centring at around 180W, owing to the gradual decay in power consumption over each operating cycle. As the Gaussian distribution cannot model skewness, it is not surprising to see that a number of points at around the 180W mark in Figure 3.9a deviate slightly from the $45^\circ$ line. Nevertheless, as far as the learning of model parameters is concerned, the results presented thus far are able to validate the advantage of using the robust EM algorithm over the non-robust variant.

### 3.4.2 Parameter Estimation for the Temporal Model

The process of estimating $\boldsymbol{\lambda}_{\mathrm{d}}$ consists of two main steps, the first of which is the extraction of appliance state durations from the appliance-level contributions in a training dataset. Also part of this outcome are the learned parameters related to the Markov state transitions. Then, in the second step, by using the extracted durations, the parameters governing the state duration distributions in (3.17) are inferred. A description of these steps is provided in the discussion that follows.

#### Extraction of Appliance State Durations

A prerequisite to the extraction of the list of sojourn times corresponding to state $i$ of appliance $k$ is the segmentation of the appliance measurements in terms of its possible states. In other words, given $y_{1:T,k}$ from the training data, $x_{1:T,k}$ needs to be estimated.

To achieve this without any knowledge of the underlying Markov process (i.e. $\boldsymbol{\pi}_k$ and $A_k$ are unknown), an iterative procedure known as the segmental $k$-means algorithm [JR90] is employed. While the algorithm in [JR90] also estimates the emission model parameters $(\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k)$ in addition to $x_{1:T,k}$, $\boldsymbol{\pi}_k$ and $A_k$, we fixed the estimates of $(\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k)$ to the values found using the robust EM algorithm described in Section 3.4.1. Also, because the starting value of a window of power measurements selected from the training data is less constrained than in the case of speech modelling where the vocalisation of a particular syllable has well-defined beginnings, the initial probability of entering state $i$ is less important in the context of our problem. Thus, instead of estimating $\boldsymbol{\pi}_k$ from the training data, the principle of indifference is used such that $\pi_i$ is assigned the same value of $1/M_k$ for all $i \in \mathcal{M}_k$. That is to say, the probability of entering state $i$ at the start of a sequence of measurements is uniformly distributed across all possible $M_k$ states of appliance $k$.

In each iteration of the segmental $k$-means algorithm, there are two parts. The first part is simply an application of the Viterbi algorithm described in Section 2.6.2, but with inputs from $y_{1:T,k}$, $(\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k^2)$ and the initial guess or previous estimate of $A_k$. This means the $n$th iteration of the first part yields the segmentation $x_{1:T,k}^{[n]}$. For the second part, the estimate of $A_k$ is updated with $x_{1:T,k}^{[n]}$ using

$$a_{i,j}^{[n+1]} = \frac{1/M_k + \sum_{t=2}^{T} [x_{t-1,k}^{[n]} = i \wedge x_{t,k}^{[n]} = j]}{1 + \sum_{j=1}^{M_k} \sum_{t=2}^{T} [x_{t-1,k}^{[n]} = i \wedge x_{t,k}^{[n]} = j]}, \tag{3.26}$$

where $[\cdot]$ is the Iverson bracket [Knu92] such that for proposition $\mathcal{V}$,

$$[\mathcal{V}] = \begin{cases} 1, & \text{if } \mathcal{V} \text{ is true,} \\ 0, & \text{otherwise.} \end{cases} \tag{3.27}$$

Note that the expression in (3.26) is different from that of the maximum likelihood estimation in [JR90], i.e.

$$a_{i,j}^{[n+1]} = \frac{\sum_{t=2}^{T} [x_{t-1,k}^{[n]} = i \wedge x_{t,k}^{[n]} = j]}{\sum_{j=1}^{M_k} \sum_{t=2}^{T} [x_{t-1,k}^{[n]} = i \wedge x_{t,k}^{[n]} = j]}, \tag{3.28}$$

since a uniform prior over each row of $A_k$ is imposed, as has been done by Shahrokni et al. [SDF04] for the detection of texture boundaries in computer vision.

Like any iterative algorithms described previously, the segmental $k$-means algorithm terminates when the likelihood $p(x_{1:T,k}, y_{1:T,k} \mid A_k)$ converges, upon which the estimate of $x_{1:T,k}$ and the estimate of $A_k$ are obtained.

Lastly, with the estimated state sequence, $\hat{x}_{1:T,k}$, the list of durations for each state $i \in \mathcal{M}_k$ can be easily extracted by recording the length of each block of consecutive $i$ in $\hat{x}_{1:T,k}$. If there are $S$ such blocks, then the length of each block $s$ is denoted by $d_s$, such that $\{d_s\}_{s=1}^{S}$.

### Inference of Model Parameters using Minimum Message Length

Suppose that a list of $S$ durations for a certain state $i$ of appliance $k$, $\{d_s\}_{s=1}^{S}$, has been obtained from the preceding step. The objective is then to estimate $\Theta_i = \{(m_{l,i}, \alpha_{l,i}, \beta_{l,i})\}_{l=1}^{L_i}$. However, because the number of components in the Gamma mixture model, $L_i$, is also unknown, it needs to be inferred as well. This is where the minimum message length (MML) principle comes in.

As described in the seminal paper of Wallace and Boulton [WB68], the core idea of the MML principle is that, the model which results in the shortest overall message length should be chosen, given that it is the most parsimonious one in the information-theoretic sense. The overall message consists of two parts; the first part is the encoded model, while the second part refers to the encoded data as a result of using the model specified in the first part of the message. Formally, the total message length (in bits or nats) is given as

$$I(\Theta, \mathcal{D}) = I(\Theta) + I(\mathcal{D} \mid \Theta), \tag{3.29}$$

where $I(\Theta)$ and $I(\mathcal{D} \mid \Theta)$ are the lengths of the first and second part of the overall message respectively; for our case, $\mathcal{D}$ corresponds to $\{d_s\}_{s=1}^{S}$, whereas $\Theta$ is $\Theta_i$.

In the context of mixture modelling and the problem of selecting the appropriate number of components $L_i$, the goal of minimising the total message length acts as a regulariser for model complexity. In particular, although having a model which has many components may allow for a more precise way of describing the data (e.g. higher likelihood $p(\mathcal{D} \mid \Theta)$ or equivalently, shorter message length for the data part $I(\mathcal{D} \mid \Theta)$), there are associated risks of overfitting. To account for this, the MML principle introduces an intuitive notion of penalising for large $L_i$. The basic concept is that each component's parameters require a certain number of bits to be encoded, and a large $L_i$ implies a longer length for the model part of the message. Therefore, in solving the MML problem, i.e.

$$\hat{\Theta} = \arg\min_{\Theta} I(\Theta, \mathcal{D}), \tag{3.30}$$

a trade-off is implicitly imposed, such that the choice of $\Theta$ still allows for efficient encoding of the data portion of the message, but without severely inflating the size required for the encoded model. In short, the optimal $\Theta$ is the one that provides the best compromise in terms of the overall message length.

The minimisation in (3.30), as applied to our problem, necessitates the specification of $I(\Theta_i)$ and $I(\{d_s\}_{s=1}^{S} \mid \Theta_i)$. The former is

$$\begin{aligned} I(\Theta_i) = L_i \log(2) + \frac{L_i - 1}{2} \log(S) - \frac{1}{2} \sum_{l=1}^{L_i} \log(m_{l,i}) - \log(L_i - 1)! \\ - \sum_{l=1}^{L_i} \log\left(\frac{1}{\beta_{l,i}}\right) - \sum_{l=1}^{L_i} \log\left(\frac{2}{\pi(1+\alpha_{l,i}^2)}\right) \\ + \frac{1}{2} \sum_{l=1}^{L_i} \log\left(\frac{S^2}{\beta_{l,i}^2}\left[\alpha_{l,i}\psi^{(1)}(\alpha_{l,i}) - 1\right]\right), \end{aligned} \tag{3.31}$$

where $\psi^{(u)}(\alpha_{l,i})$ denotes the $u$th order polygamma function defined as

$$\psi^{(u)}(\alpha_{l,i}) = \frac{d^{u+1}}{d\alpha_{l,i}^{u+1}} \log\left(\Gamma(\alpha_{l,i})\right), \tag{3.32}$$

while the latter is

$$I(\{d_s\}_{s=1}^{S} \mid \Theta_i) = -\sum_{s=1}^{S} \log\left(\sum_{l=1}^{L_i} m_{l,i} g(d_s; \alpha_{l,i}, \beta_{l,i})\right), \tag{3.33}$$

with $g(d_s; \alpha_{l,i}, \beta_{l,i})$ being the probability density function of the Gamma distribution. A complete derivation of these expressions is presented in Section A.1 of Appendix 1.

Then, for a given $L_i$, the optimum $\Theta_i$ in (3.30) is obtained using the EM algorithm. In the $n$th iteration, the E-step involves calculating the posterior probability of assigning the $s$th duration to each mixture component $l$,

$$r_{ls}^{[n]} = \frac{m_{l,i}^{[n]} g(d_s; \alpha_{l,i}^{[n]}, \beta_{l,i}^{[n]})}{\sum_{l=1}^{L_i} m_{l,i}^{[n]} g(d_s; \alpha_{l,i}^{[n]}, \beta_{l,i}^{[n]})}. \tag{3.34}$$

This is followed by the M-step, whereby the parameter estimates are updated by computing

$$m_{l,i}^{[n+1]} = \frac{\frac{1}{2} + S_l^{[n]}}{S + \frac{L_i}{2}} \tag{3.35}$$

$$\beta_{l,i}^{[n+1]} = \frac{\sum_{s=1}^{S} d_s r_{ls}^{[n]}}{\alpha_{l,i}^{[n]} S_l^{[n]}}, \tag{3.36}$$

where $S_l^{[n]} = \sum_{s=1}^{S} r_{ls}^{[n]}$. Unfortunately, the expression for $\alpha_{l,i}^{[n+1]}$ does not have a closed form, given that it is defined implicitly by

$$\begin{aligned} \log\left(\frac{\sum_{s=1}^{S} d_s r_{ls}^{[n]}}{S_l^{[n]}}\right) - \frac{\sum_{s=1}^{S} r_{ls}^{[n]} \log(d_s)}{S_l^{[n]}} + \frac{2\alpha_{l,i}^{[n+1]}}{S_l^{[n]}\left[1 + \left(\alpha_{l,i}^{[n+1]}\right)^2\right]} \\ + \frac{1}{2S_l^{[n]}}\left(\frac{\alpha_{l,i}^{[n+1]}\psi^{(2)}(\alpha_{l,i}^{[n+1]}) + \psi^{(1)}(\alpha_{l,i}^{[n+1]})}{\alpha_{l,i}^{[n+1]}\psi^{(1)}(\alpha_{l,i}^{[n+1]}) - 1}\right) \\ - \log(\alpha_{l,i}^{[n+1]}) + \psi^{(0)}(\alpha_{l,i}^{[n+1]}) = 0. \end{aligned} \tag{3.37}$$

As such, $\alpha_{l,i}^{[n+1]}$ is searched for numerically using root-finding algorithms. For a complete derivation of these equations, see Section A.2 of Appendix 1.

Over multiple iterations, the E-step and M-step are repeated until the total message length $I(\Theta_i, \{d_s\}_{s=1}^{S})$ converges, at which point, the estimated $\Theta_i$ and the corresponding total message length as a result of using $L_i$ components are recorded. Then, with multiple runs of the EM algorithm for a range of different $L_i$, the $\Theta_i$ with the $L_i$ which results in the shortest message length is selected as the learned model.

**Examples of applying to real-world appliance data**

Figure 3.11 shows the variation in message length as the number of mixture components increases, when learning is performed on the extracted durations of the OFF state from the dishwasher shown in Figure 3.4. It illustrates that, for this particular case, two mixture components should be used as the smallest $I(\mathcal{D}, \Theta)$ occurs at $L_i = 2$.

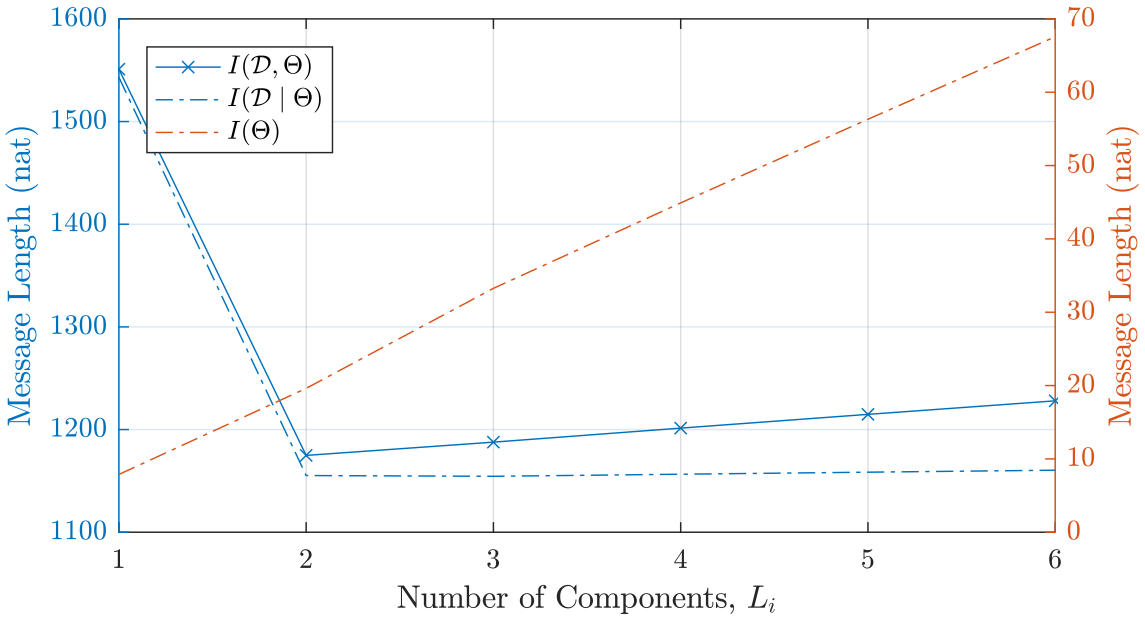



Figure 3.11: The variation in message length with the number of mixture components $L_i$. Learning is performed on the extracted durations of the OFF state from the dishwasher shown in Figure 3.4.

The result is consistent with the description given in Section 3.3.2 on the OFF state duration of the dishwasher. One component is for modelling the short OFF periods embedded in an operating cycle while the other component is responsible for modelling the long inter-usage durations. Thus, this outcome of using MML does not deviate from the prior expectation that two mixture components should be used. Also shown is the fitted mixture of two Gamma distributions in Figure 3.12.

### 3.4.3 Summary

A summary of the learning process as described in the previous two subsections is given in Figure 3.13.

For each appliance $k$, the power measurements from the training data are used for estimating the mean and variance of each state via the robust EM algorithm.

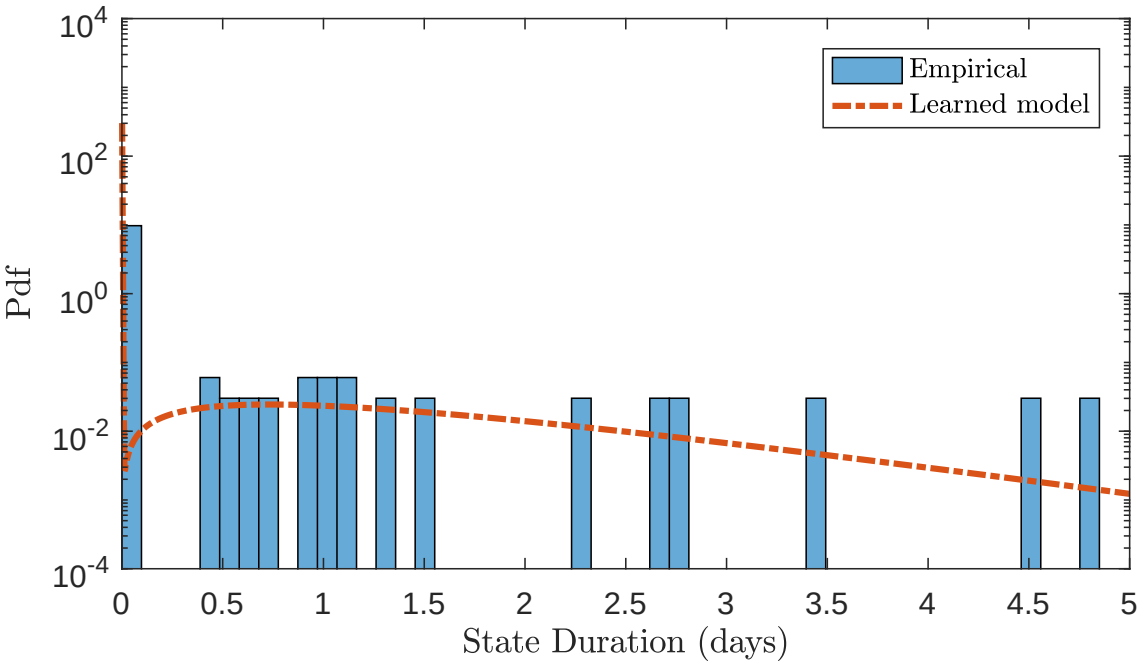



Figure 3.12: The learned model of the OFF state duration of the dishwasher shown in Figure 3.4.

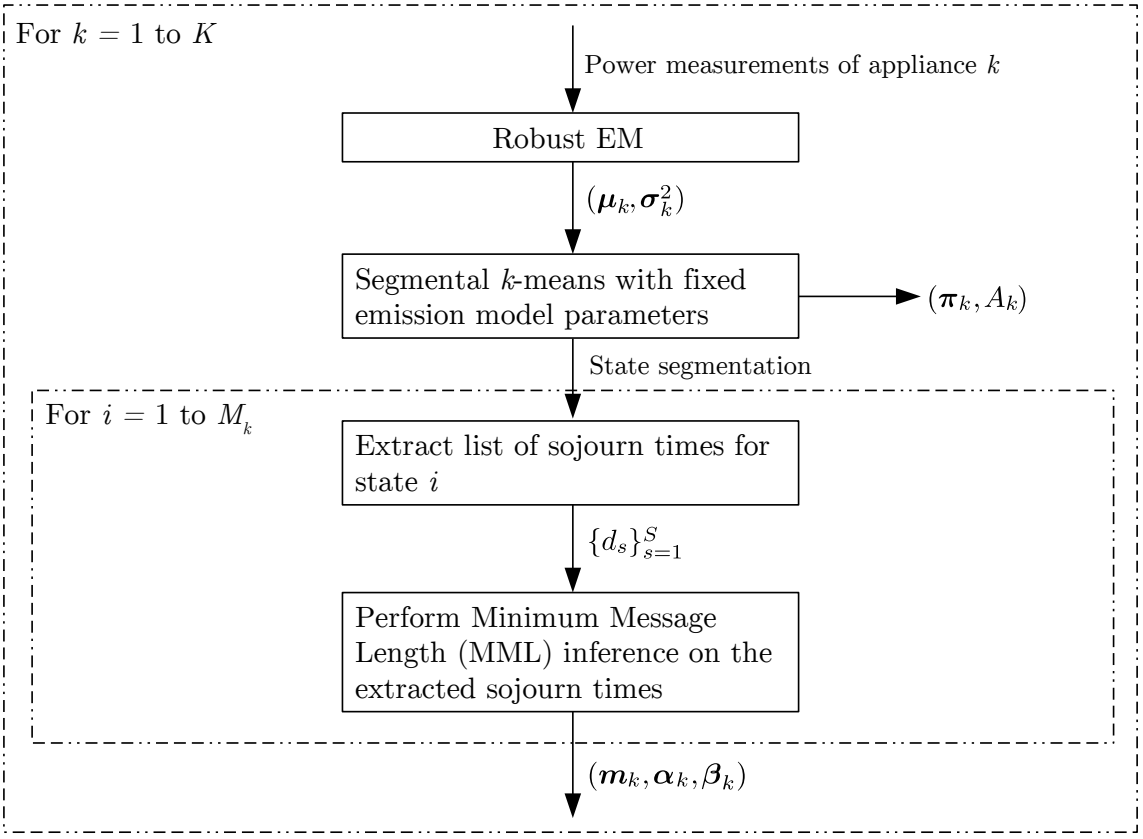



Figure 3.13: Overview of the learning procedure

Next, using the segmental $k$-means algorithm, the initial state probability and the Markov state transition matrix of appliance $k$ are estimated, while the states corresponding to the power measurements are inferred. Then, a list of durations for state $i$ of appliance $k$ is extracted, from which the parameters of a mixture of Gamma distributions are estimated by means of the MML principle. Finally, the learning process terminates when the parameters for each state and each appliance are obtained, i.e. $\boldsymbol{\lambda}_{\mathrm{e}}$ and $\boldsymbol{\lambda}_{\mathrm{d}}$.

## 3.5 Experimental Results and Discussion

In this section, we validate the proposed model, FVTHMM, in terms of its ability to generate power samples that closely mirror the operational behaviour of modelled appliances, and its ability in aiding the separation of appliances with similar power consumption during disaggregation.

### 3.5.1 Generation of Appliance Power Consumption

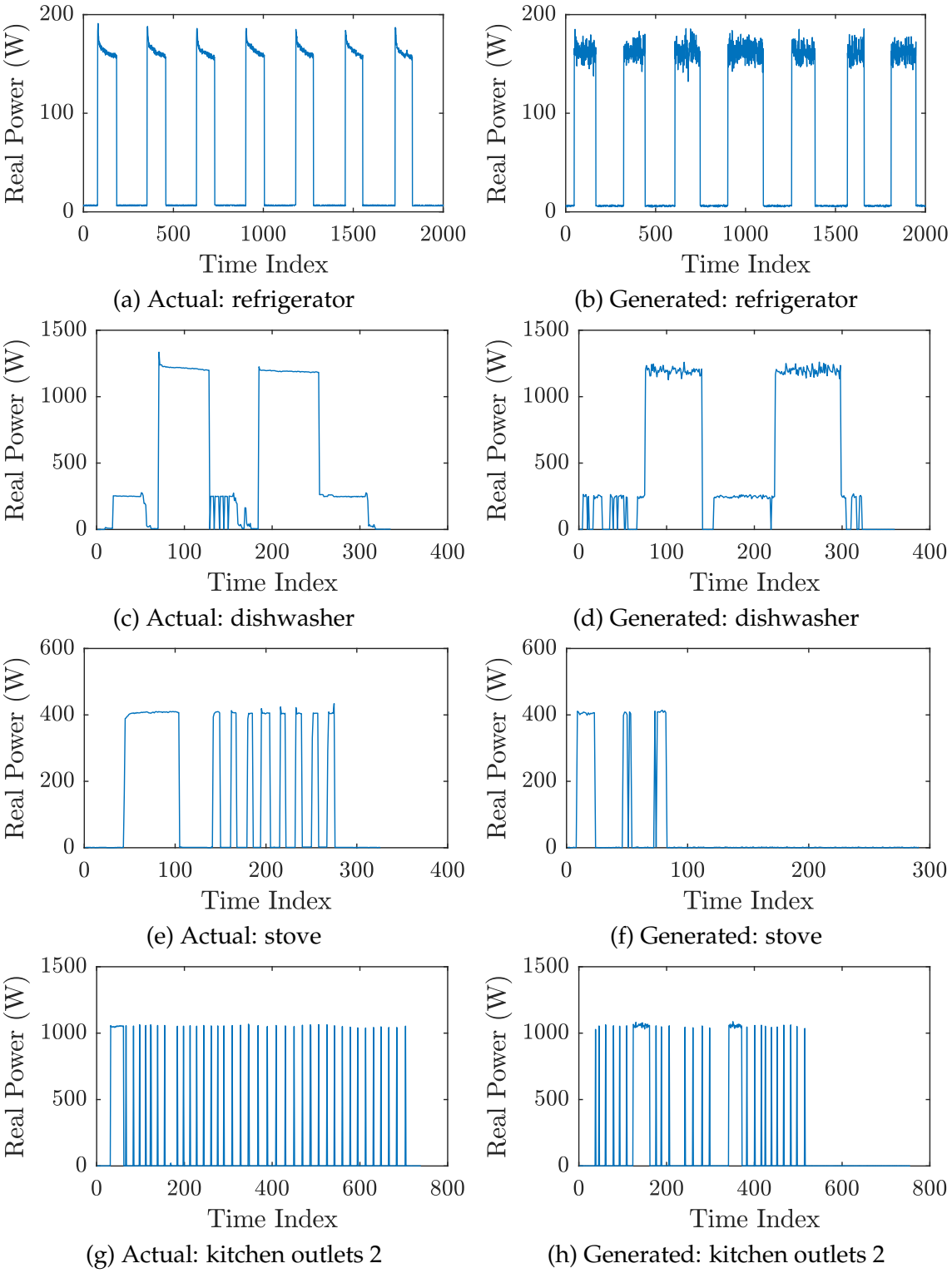
Although the end goal of this research is to produce estimates of power consumption for each appliance from the aggregate power measurements, it is interesting to consider the reverse process in which power consumption measurements of appliances are generated from the learned models, for validation purposes. This is possible as the model is that of a stochastic process, of which realisations can be made. To that end, power samples are drawn probabilistically and their outputs are compared against the actual consumption data from the training set. For demonstration purposes, we have chosen to use appliances in house 2 from the REDD dataset as the basis for this comparison.

Among the important loads considered are the refrigerator, the dishwasher, the stove and the kitchen outlets. Their power consumption and the generated counterparts are presented in Figure 3.14. A number of observations can be made. Firstly, the generated power values of the refrigerator and the dishwasher, while they are operating, have noticeably larger variance than that of the actual power values. This is attributed to the non-stationarity of the power consumption data as a result of the gradual decay in power that is apparent in both cases. Because the emission model is assumed to be stationary, the variance of the fitted distribution is inflated. Therefore, it is to be expected that the generated data has a larger noise level. A solution to this is to relax the assumption, such that the power consumption for a given state is no longer i.i.d. across time, but dependent on

the duration from which the state is first entered, i.e. $p(y_{t,k} \mid x_{t,k}, c_{t,k})$ instead of $p(y_{t,k} \mid x_{t,k})$ in (3.15). A brief investigation of such an approach is described in Section 4.5 of Chapter 4.

Secondly, the figure also illustrates that, for the dishwasher and the kitchen outlets 2, the time progression of the generated power values does not seem to follow that of the actual power values. Among other apparent differences, the rapid-switching of power consumption for the dishwasher in the middle of its operating cycle does not occur for the generated version. Likewise, the wide pulse from the actual measurements of the kitchen outlets 2 always happens at the start of each operating cycle, but it is not reflected in its generated counterpart. Instead, two wide pulses are located in the middle, at least for the realisation shown in Figure 3.14h. This deviation from actual behaviour is a result of associating a single state to each power level. In particular, one state of the dishwasher is tied to the cluster of power values centring at around 250W, regardless of where in the operating cycle 250W is consumed. This notion of a state does not reflect the device's actual behaviour, as the first 250W of the dishwasher could be related to the "wash" state while subsequent ones might belong to other device's states. The same can also be said for the one-to-one mapping between one state of the kitchen outlets 2 and the power draw of 1000W. If additional states could be introduced to distinguish between similar power levels occurring at different times in one device activation, power values closely mirroring that of the actual behaviour of appliances could be generated. However, because our objective is not to build an accurate appliance simulator, the one-to-one mapping is maintained for the task of disaggregation as described in subsequent chapters. In fact, this notion of a state appears to be employed in majority of existing work on NILM [MHHE11, KJ11, EBE15, MPB+16, KDM+16].

Lastly, one other observation is the clear difference between the actual and the generated power values for the stove, as shown in Figures 3.14e and 3.14f. The main reason for this, in addition to the one-to-one mapping described earlier, is the use of only a single Gamma distribution for representing the state duration of the stove, since there is a lack of duration data for fitting a more complex model with many components; the stove in house 2 of the REDD dataset was only used twice during the monitoring period. As such, there is insufficient data, and samples drawn from a single Gamma distribution is not likely to replicate the distinctive pulses in the actual power values of the stove.




(a) Actual: refrigerator

(b) Generated: refrigerator

(c) Actual: dishwasher

(d) Generated: dishwasher

(e) Actual: stove

(f) Generated: stove

(g) Actual: kitchen outlets 2

(h) Generated: kitchen outlets 2

Figure 3.14: Comparison between the actual power consumption and the generated power consumption for different appliances in house 2 of the REDD dataset.

Nevertheless, in spite of the limitation as an appliance simulator given the definition of "state" used, the ability to incorporate state durations for the purpose of load disaggregation is invaluable, as we shall see in the next subsection.

### 3.5.2 Robustness Against Overlaps in Power Features

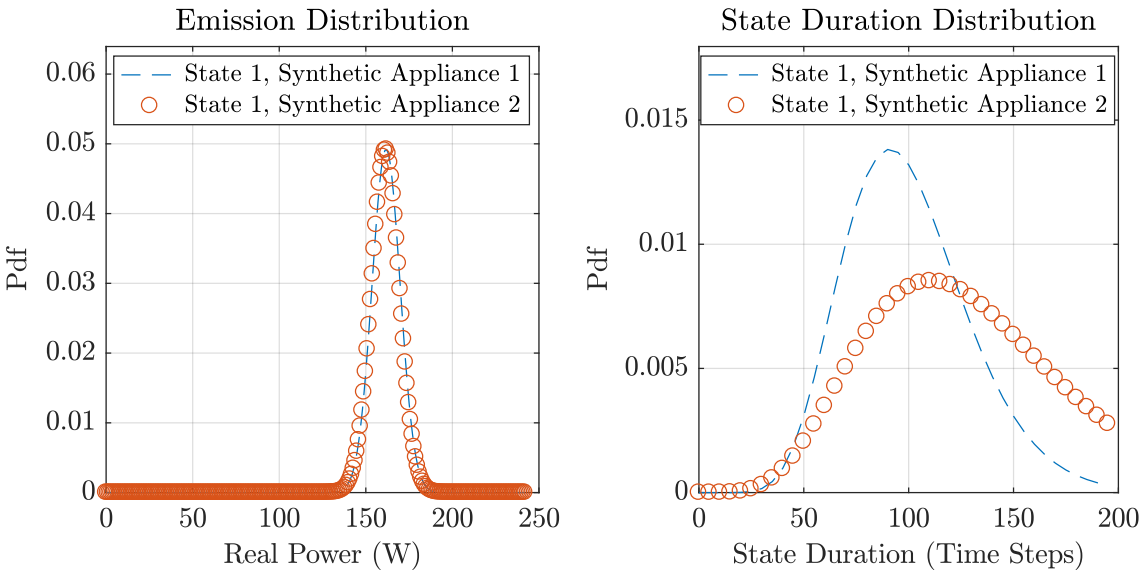
As mentioned in Chapter 2, one of the most important issues in the disaggregation of low sampling rate data is the overlaps or similarities in power features between appliances. To validate the robustness of FVTHMM in this regard, the two sequences of power consumption data are generated, each corresponding to a synthetic appliance. For testing against the worst case scenario, both synthetic appliances are configured to have the same power consumption but different state duration characteristics. The two sequences are then added together to form the aggregate data. Relative to FHMM, comparison is made on the basis of how well FVTHMM enables the two synthetic appliances to be identified correctly.

The emission model and the state duration model of the synthetic appliances are specified by random variables that have a Gaussian distribution and a Gamma distribution respectively. Figure 3.15 shows the model for the ON state of the synthetic appliances. The full model specification is summarised in Table 3.1 and Table 3.2. For disaggregation with FVTHMM, the parameters used are exactly the same as those employed for generating synthetic data, while for the case of FHMM, the Markov state transition matrices used have self-transition probabilities that are consistent with the mean state durations, i.e. $a_{i,i} = \frac{E[d]-1}{E[d]}$. For example, the mean duration of the ON state of appliance 1 is 100 time steps. Therefore, $a_{i,i}$ for $i = 1$ is 0.99. Table 3.3 shows the complete Markov state transition matrices derived in this manner.

Table 3.1: Emission model of the synthetic appliances.

| **Synthetic Appliance** | **State,** $x_{t,k}$ | **Mean,** $\mu$ | **Standard Deviation,** $\sigma$ |
|---|---|---|---|
| 1 | 0 | 6.053 | 0.454 |
| | 1 | 161.713 | 8.105 |
| 2 | 0 | 6.053 | 0.454 |
| | 1 | 161.713 | 8.105 |

Given that the number of appliances considered for this test is small, the Viterbi algorithm is used for state inference under both FVTHMM and FHMM. For real-world situations with many appliances however, this is no longer com-




(a) Power consumption (b) State duration

Figure 3.15: The probability density function over power consumption and state duration for the ON state of the synthetic appliances.

Table 3.2: State duration model of the synthetic appliances.

| **Synthetic Appliance** | **State,** $x_{t,k}$ | **Shape,** $\alpha$ | **Scale,** $\beta$ |
|---|---|---|---|
| 1 | 0 | 100.000 | 2.000 |
| | 1 | 11.111 | 9.000 |
| 2 | 0 | 40.111 | 4.737 |
| | 1 | 6.760 | 19.231 |

Table 3.3: State transition matrices used for disaggregation under FHMM.

(a) Synthetic appliance 1

| **State,** $x_{t,1}$ | **0** | **1** |
|---|---|---|
| **0** | 0.9950 | 0.0050 |
| **1** | 0.0100 | 0.9900 |

(b) Synthetic appliance 2

| **State,** $x_{t,2}$ | **0** | **1** |
|---|---|---|
| **0** | 0.9497 | 0.0503 |
| **1** | 0.0077 | 0.9923 |

putationally tractable. As such, a new algorithm is developed and it is detailed in Chapter 4.

Figure 3.16 presents the disaggregation outcome on one instance of the generated synthetic data shown in Figure 3.16a. Compared to FHMM, it is apparent that FVTHMM is able to reconstruct the contributions of appliance 1 and appliance 2 accurately, even when there is essentially no difference in power consumption between the two appliances. In contrast, FHMM fails to give correct results as the state transition information is not sufficient to provide the means to

disambiguate between appliance 1 and appliance 2. Indeed, the state transition probabilities in Table 3.3 are similar, despite actual differences in state duration characteristics.

Overall, the results presented herein confirm the advantage of modelling the state duration explicitly as demonstrated in the work by Kim et al. [KAL11] and Johnson and Willsky [JW13]. Though, with FVTHMM, the state transition probability can be updated incrementally and dynamically, facilitating the implementation of a real-time load disaggregation system. This is exemplified in Figure 3.17, where the hazard function values for the corresponding estimates given in Figure 3.16b are shown. It illustrates that the probability of switching states is not static but increases with the number of time steps since entering the current state.

## 3.6 Summary

In this chapter, we have presented an alternative variant of the hidden semi-Markov model (HSMM) for representing appliance behaviour: factorial variable transition hidden Markov model (FVTHMM). Besides being able to explicitly account for the duration of states, it also incorporates the notion of time-varying duration-dependent state transition probability. This allows for the real-time updates of probability values during state inference.

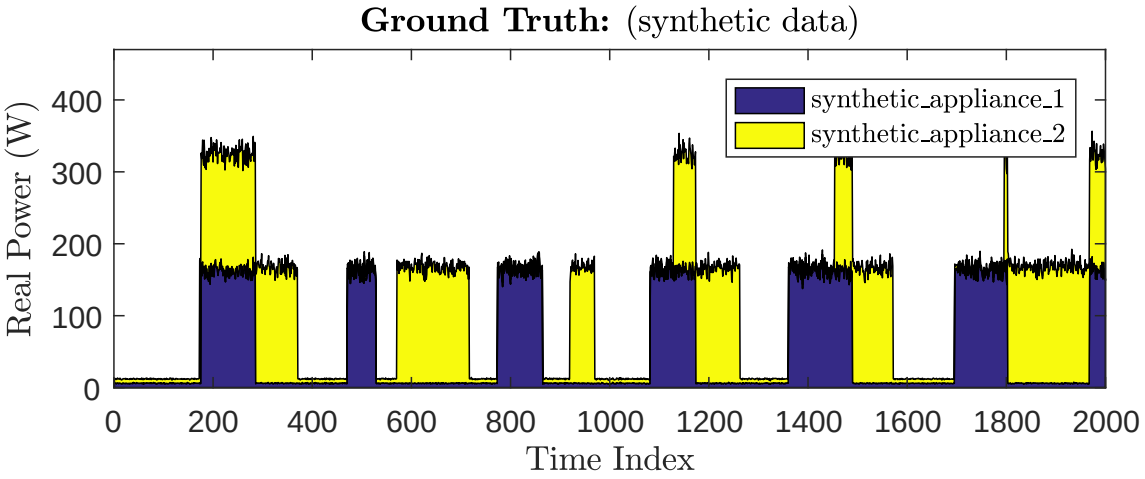
Further, we attempted to generate power values from the learned appliance model. While the outcome does not exactly mirror that of the actual behaviour of appliances, the discrepancy is not intrinsic to FVTHMM. Instead, it stems from the one-to-one association between the states and power levels.

Additionally, we showed that compared to established methods based on FHMM, overlaps in power consumption between appliances could be resolved with our proposed model, confirming the significance of modelling state durations when only low rate power consumption measurements are available.

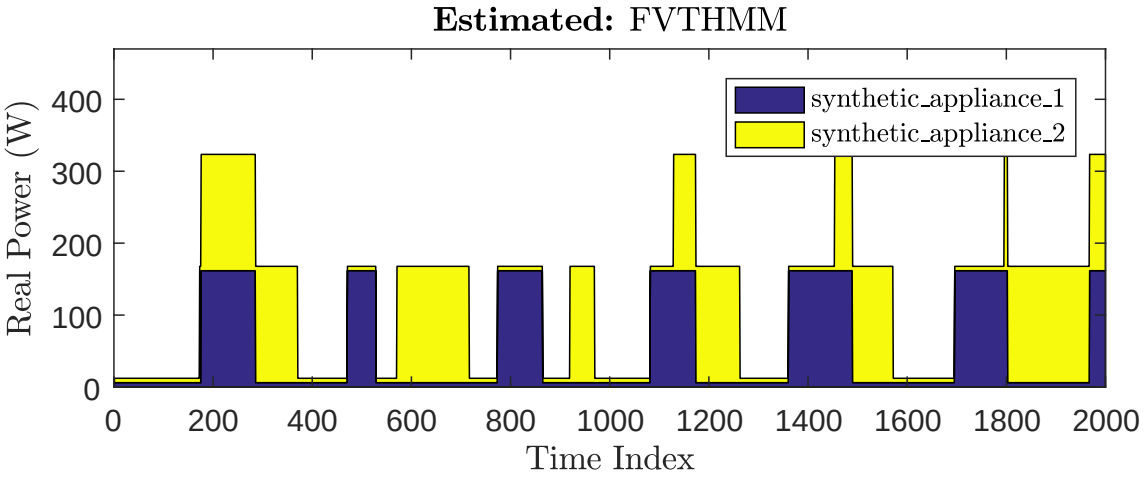
For learning model parameters, the use of a robust EM algorithm is demonstrated. In comparison to an ordinary EM algorithm, it was found that outlying values due to transients and other anomalies are less likely to affect the fitted distribution.

Lastly, the minimum message length (MML) principle has not been used in NILM before and it was shown to be valuable in automatically inferring the number of clusters inherent in the duration distribution. Other issues which have not been discussed in this chapter are the specifics of efficient inference of states un-

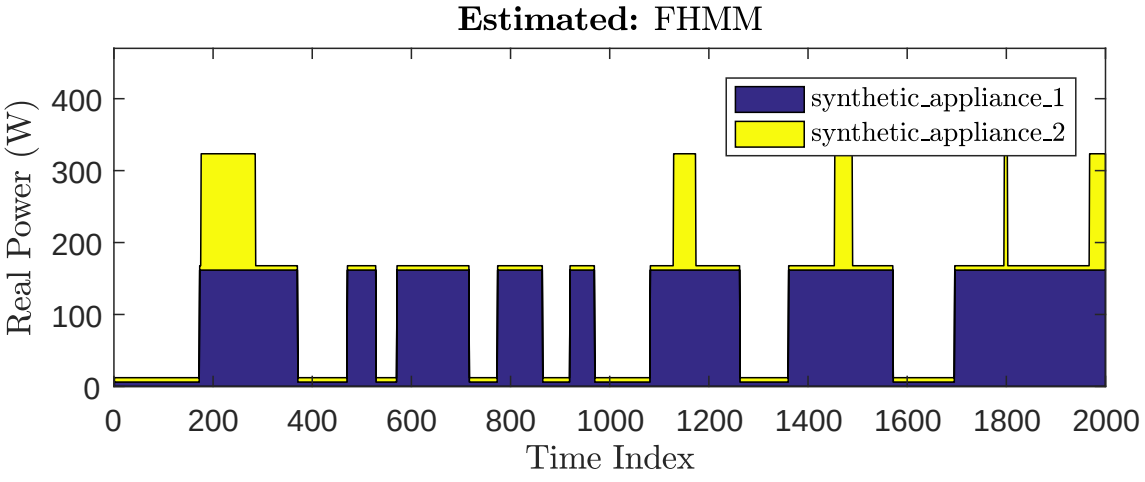
der FVTHMM and the computational scalability for cases with many appliances. Detailed discussion on these are given in the next chapter.




(a) The generated synthetic data




(b) Disaggregation with FVTHMM




(c) Disaggregation with FHMM

Figure 3.16: Comparison between the ability of FVTHMM and FHMM in identifying two synthetic appliances with the same power consumption but different state duration characteristics.

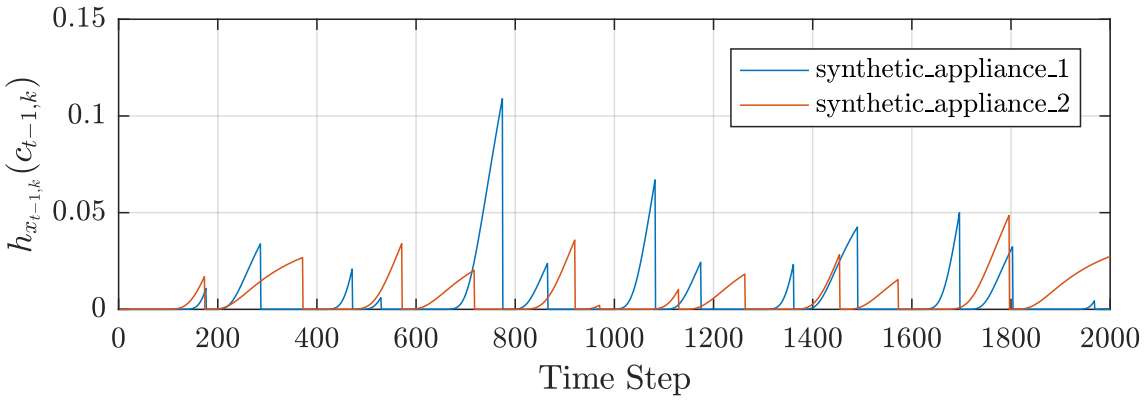



Figure 3.17: The time progression of the hazard function or the probability of switching states as used by FVTHMM.

CHAPTER FOUR

# APPLIANCE STATE INFERENCE

Appliance state inference plays a central role in Non-intrusive Load Monitoring (NILM). The objective is to estimate the states of each appliance of interest given the observed aggregate measurements. Although there are various techniques that could be used for state inference under the proposed factorial variable transition hidden Markov model (FVTHMM), most do not meet the objective of performing disaggregation in real-time and do not scale well computationally. Thus, a new algorithm – Particle-Based Distribution Truncation (PBDT) – is proposed for overcoming such limitations. It combines the dynamic programming approach of the Viterbi algorithm and the survival-of-the-fittest concept from particle filters, allowing multiple appliance state trajectories to be tracked in real-time efficiently. In this chapter, we begin by providing the motivation for the proposed method, before describing the computational issues of the Viterbi algorithm when applied to FVTHMM. Then, in addressing the pertinent limitations, a comprehensive account of the PBDT algorithm is given, with emphasis on certain properties that could be exploited to facilitate the sharing of computation results for improving computational performance. This is followed by a detailed evaluation of its disaggregation accuracy over the data from real homes, and the validation of its time complexity in relation to real-time applications. Also introduced is a new metric for identifying the source of disaggregation errors. Lastly, the incorporation of features based on power decays into FVTHMM is briefly investigated and its improvements are presented. Part of the work pertaining to this chapter has been published in a journal paper [WcD14].

## 4.1 Introduction and Related Work

The proposed model, FVTHMM, as described in Chapter 3, is an instance of a broader class of latent variable models. The internal state of appliance $k$ at time

$t$, $x_{t,k}$, is considered hidden, and accordingly, determines its power consumption, $y_{t,k}$. As such, inferring the unknown states of appliances given the aggregate-level measurements is the main objective to be pursued, as far as load disaggregation is concerned. One standard way of estimating the states is to select those for which the likelihood of observing the aggregate-level measurements is maximised, or formally,

$$\hat{\mathbf{x}}_{1:T} = \arg\max_{\mathbf{x}_{1:T}} p(\mathbf{x}_{1:T}, y_{1:T}), \tag{4.1}$$

where $\mathbf{x}_t$ refers to the system state consisting of the states of $K$ appliances at time $t$, i.e. $\mathbf{x}_t = (x_{t,1}, \ldots, x_{t,K})$, and $y_t$ is the corresponding aggregate power consumption.

The maximisation in (4.1) is typically performed using the Viterbi algorithm. However, due to the extended state space induced by the additional counter variables (see (3.2)), its direct application under FVTHMM is not computationally feasible, except for a limited set of impractical scenarios. While other techniques such as simulated annealing and Gibbs sampling could be adapted from previous work [KAL11, JW13], they are only inherently suited for batch-processing; blocks of aggregate measurements have to be obtained (e.g. a day's worth) before inference of states is performed retrospectively. Therefore, as mentioned in Section 2.6.2 of Chapter 2, these approaches are limited in their ability to disaggregate power consumption measurements in real-time, thereby preventing NILM from offering low-latency feedbacks required as part of increasing user engagement towards conserving energy [PSJ+07] and limiting it from being used as the basis for interactive applications noted in Chapter 1.

To that end, a new state inference algorithm is clearly needed to meet the goal of this research. The main contributions presented in this chapter are:

- A computationally efficient algorithm for the real-time tracking of appliance states. Sharing of computation results and intelligent pruning of implausible solutions enable the method to be run on houses with a combined system state count of 20 billion.

- A tractable method for inferring the hidden states under the proposed factorial variable transition hidden Markov model (FVTHMM).

- A metric for quantifying whether the errors in state inferences are due to modelling inaccuracies or an artefact of the incorrect pruning of solutions.

- Improvements to the base FVTHMM model presented in Chapter 3 for taking into account gradual decays in power consumption.

Note that, the state inference algorithm presented in this chapter has been completed in 2014. While several NILM approaches [MPB+16, IS16, KDM+16, KDH+16, TWLT16] have been developed since then (see Chapter 2), none were particularly suitable for our probabilistic model detailed in Chapter 3. One approach, published in late 2016 by Lange and Bergés [LB16], however, followed our published work of 2014 [WcD14], giving credence to the method presented here. In particular, their state inference algorithm is based on pruning improbable solutions, though unlike our earlier work, the model to which their method is applied does not capture state durations explicitly. This has potential implications in differentiating between appliances with similar power consumption. Moreover, the method which we developed is more essential under our more powerful model. Without it, computations can be intractable, as we shall see in the subsequent sections.

## 4.2 Computational Issues of the Viterbi Algorithm

We have shown in Section 2.6.2 of Chapter 2 the description of the Viterbi algorithm for an ordinary hidden Markov model (HMM). In this section, we expand on that discussion for the VTHMM and the FVTHMM from an algorithmic and computational perspective, to highlight the issues of computational intractability and to motivate the need for the developed approach as detailed in Section 4.3.

### 4.2.1 Complexity Analysis Under VTHMM

Recall from Chapter 3 that a defining part of the VTHMM formulation is the time-varying duration-dependent state transition probability. The joint probability over all its random variables is

$$\begin{aligned} p(x_{1:T}, y_{1:T}, c_{1:T}) = p(x_1)p(c_1) \prod_{t=1}^{T} p(y_t \mid x_t) \\ \times \prod_{r=2}^{T} p(x_r \mid x_{r-1}, c_{r-1}) p(c_t \mid x_r, c_{r-1}, x_{r-1}), \end{aligned} \tag{4.2}$$

where $x_t$ is the state at time $t$, $c_t$ is the corresponding counter value and $y_t$ is the observed value at time $t$.

In maximising the joint probability over the space of possible $x_{1:T}$ and $c_{1:T}$, the base case (i.e. $t = 1$) and the general case of the Viterbi score $\delta_t(\cdot)$ are

$$\delta_1(i, c_1) = \begin{cases} p(x_1 = i)p(y_1 \mid x_1 = i), & \text{if } c_1 = 1 \\ 0, & \text{otherwise} \end{cases} \tag{4.3}$$

$$\delta_t(j, c_t) = \begin{cases} \max\limits_{\substack{1 \le i \le M \\ i \neq j}} \Big[ p(y_t \mid x_t = j) \max\limits_{1 \le \tau \le t-1} \Big( \delta_{t-1}(i, \tau) \\ \qquad \times p(x_t = j \mid x_{t-1} = i, c_{t-1} = \tau) \Big) \Big], & \text{if } c_t = 1 \\ \delta_{t-1}(j, \tau) p(y_t \mid x_t = j) \\ \times p(x_t = j \mid x_{t-1} = j, c_{t-1} = \tau), & \text{otherwise.} \end{cases} \tag{4.4}$$

respectively [RW92]. Note that the factor $p(c_t \mid x_t, c_{t-1}, x_{t-1})$ is not present in (4.4) since it is implicitly taken into account by the conditioning on $c_t$ in the piece-wise expression.

The two conditions in (4.4) are required to maintain the consistency in the relationship between the counters and any state changes; the counters must reset to one whenever a state change occur while the counters have to be incremented by one if the state remains the same as that of the previous time step. As such, for a given augmented state $(j, \tau)$ with device state $j$ and a counter value of $\tau \neq 1$ at time $t$, the only possible originating augmented state at time $t-1$ is $(j, \tau-1)$; all the other transitions to $(j, \tau)$ are impossible by construction. In contrast, a destination augmented state with $\tau = 1$ for any $j$ can be a result of transitioning from an augmented states with any counter value up to $t - 1$ as long as the previous device state was $i \neq j$.

Although these constraints show that not all transitions need to be explicitly evaluated in the Viterbi algorithm of VTHMM, the upper bound for the number of required computations is still large; if the state cardinality is $M$ and there are $T$ observations, the time complexity for inferring the hidden states is $O(M^2T^3)$. To understand this, consider the trellis structure for a 2-state VTHMM (i.e. $M = 2$) shown in Figure 4.1. It illustrates that the number of augmented state starts out at $M$ and grows by an additional $M$ at each time step before finally reaching $MT$ at time $T$. Hence, a direct application of the Viterbi algorithm would require $M + M^2 \sum_{t=2}^{T} t(t-1)$ operations in total, culminating in a time complexity of $O(M^2T^3)$. While the Viterbi algorithm in this instance is tractable for reasonable

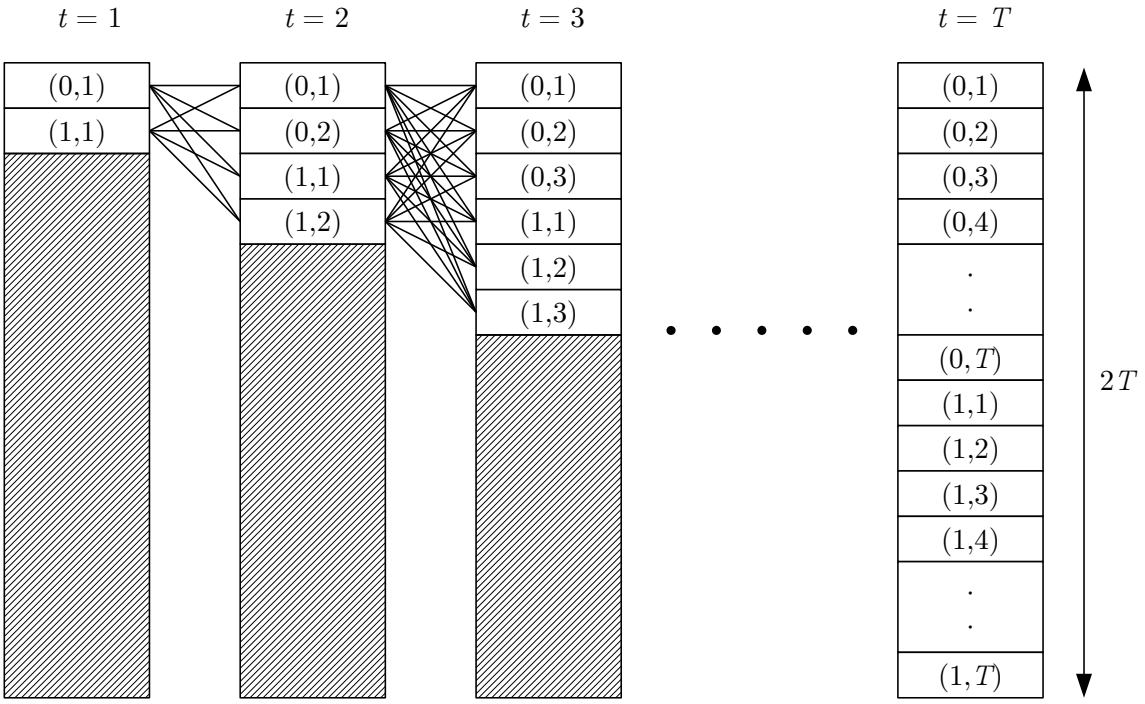



Figure 4.1: Trellis structure corresponding to a 2-state VTHMM. The notation used in each cell is $(x_t, c_t)$.

values of $M$ and $T$, its use for the factorial variant of VTHMM is computationally difficult, as we shall see in the next subsection.

### 4.2.2 Complexity Analysis Under FVTHMM

The formulation of the Viterbi algorithm for a $K$-chain FVTHMM is a straightforward extension from that of the VTHMM. If we let $i_k$ and $j_k$ denote the previous state and the current state of the $k$th chain respectively, and if we also define a set of indices, $G$, such that $G = \{k \in \{1, \ldots, K\} \mid i_k = j_k\}$, the base case and the general case of the Viterbi recursion can be written as

$$\delta_1(\mathbf{i}, \mathbf{c}_1) = \begin{cases} p(\mathbf{x}_1 = \mathbf{i}) p(y_1 \mid \mathbf{x}_1 = \mathbf{i}), & \text{if } \mathbf{c}_1 = \mathbf{1} \\ 0, & \text{otherwise,} \end{cases} \tag{4.5}$$

$$\begin{aligned} \delta_t(\mathbf{j}, \boldsymbol{\tau}^+) = \max_{\mathbf{i}, \boldsymbol{\tau}} \Big[ & \delta_{t-1}(\mathbf{i}, \boldsymbol{\tau}) p(y_t \mid \mathbf{x}_t = \mathbf{j}) \\ & \times p(\mathbf{x}_t = \mathbf{j} \mid \mathbf{x}_{t-1} = \mathbf{i}, \mathbf{c}_{t-1} = \boldsymbol{\tau}) \\ & \times p(\mathbf{c}_t = \boldsymbol{\tau}^+ \mid \mathbf{x}_t = \mathbf{j}, \mathbf{c}_{t-1} = \boldsymbol{\tau}, \mathbf{x}_{t-1} = \mathbf{i}) \Big], \end{aligned} \tag{4.6}$$

where $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_K)$, $\mathbf{i} = (i_1, \ldots, i_K)$, $\mathbf{j} = (j_1, \ldots, j_K)$, $\mathbf{1}$ is a vector of 1s of size $K$ and $\boldsymbol{\tau}^+$ is a vector whose $k$th element is $\tau_k + 1$ if $k \in G$ and 1 otherwise. For transitions that do not result in any state changes (i.e. $\mathbf{i} = \mathbf{j}$), (4.6) simplifies to

$$\begin{aligned}\delta_t(\mathbf{i}, \boldsymbol{\tau} + \mathbf{1}) = \delta_{t-1}(\mathbf{i}, \boldsymbol{\tau}) p(y_t \mid \mathbf{x}_t = \mathbf{i}) \\ \times p(\mathbf{x}_t = \mathbf{i} \mid \mathbf{x}_{t-1} = \mathbf{i}, \mathbf{c}_{t-1} = \boldsymbol{\tau}),\end{aligned} \tag{4.7}$$

mirroring the second case of (4.4).

Figure 4.2 shows the trellis structure of the Viterbi algorithm for a FVTHMM with two 2-state chains (i.e. $K = 2$, $M = 2$). Compared to before, the number of cells at each time step $t$ is now $M^K t^K$, growing to $M^K T^K$ at the end, while there are $M^K + M^{2K} \sum_{t=2}^{T} t^K (t-1)^K$ operations in total for the Viterbi algorithm. This means, the space complexity and the time complexity are $O(M^K T^{K+1})$ and $O(M^{2K} T^{2K+1})$ respectively. To get a sense of this, consider ten 2-state appliances and a day's worth of aggregate power data obtained through a sampling rate of 1Hz (i.e. $K = 10$, $M = 2$ and $T = 86400$). The number of computations needed would be in the order of $10^{214}$, which is a factor gain of $O(T^{2K}) = 10^{197}$ relative to a FHMM with the same $K$, $M$ and $T$. Clearly, coupled with the exponential growth

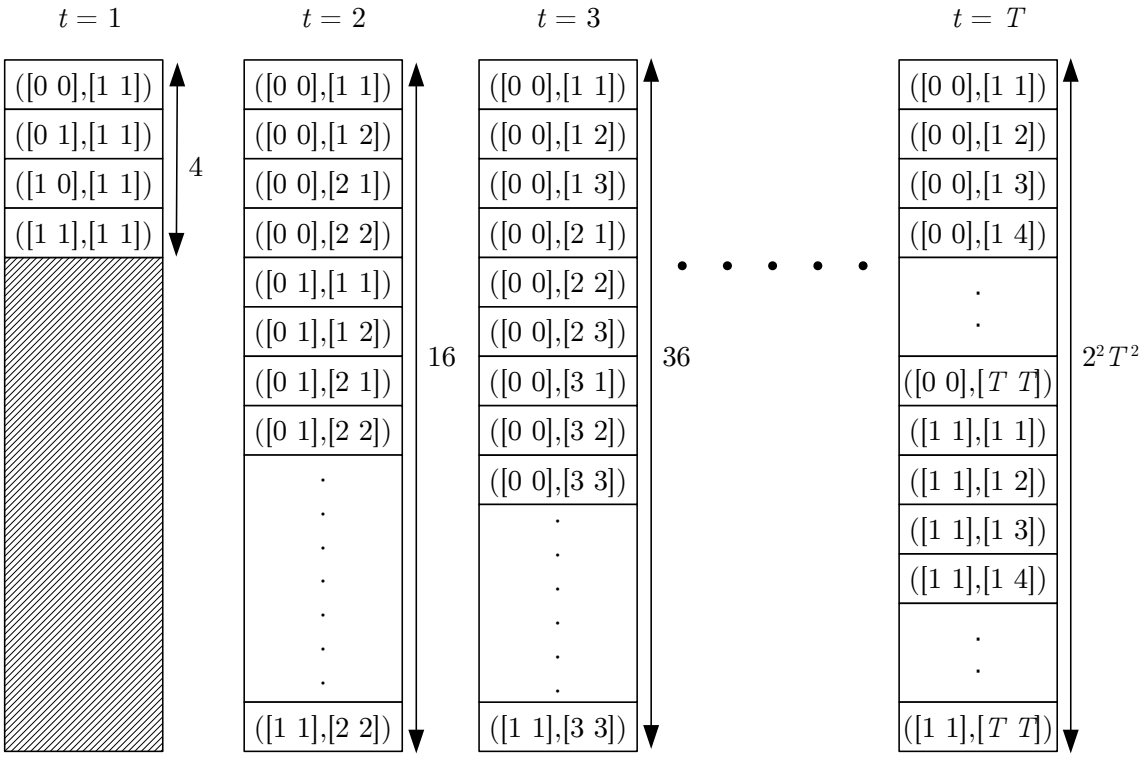



Figure 4.2: Trellis structure corresponding to a FVTHMM model with two 2-state chains (i.e. $M = 2, K = 2$). The notation used in each cell is $(\mathbf{x}_t, \mathbf{c}_t)$. To prevent clutter, the lines illustrating the trellis connections have been removed.

in the space required to store the trellis structure, the use of Viterbi algorithm for exact state inference under FVTHMM is not tractable.

## 4.3 Particle-Based Distribution Truncation (PBDT)

To address the computational issues pertaining to state inferences in FVTHMM, we proposed a new method named Particle-Based Distribution Truncation (PBDT). The method is an approximation to the Viterbi algorithm, combining the dynamic programming approach of the latter with the survival-of-the-fittest concept from particle filters.

In the subsections that follow, a detailed description of the algorithm will be first given. Then, we present further computational optimisations that should be included as part of an implementation. This is followed by a comparison between the PBDT algorithm and the Viterbi algorithm.

### 4.3.1 Algorithm

Central to the PBDT algorithm is the notion of particles. Each $n$th particle is a data structure carrying a set of relevant attributes. The attributes are a hypothesis or a state estimate $\mathcal{X}_t(n)$ for explaining the observed measurement at time $t$ (e.g. aggregate power measurement at time $t$), an associated score $\mathcal{S}_t(n)$ (e.g. the likelihood of a state sequence ending with $\mathcal{X}_t(n)$), and a reference to a parent particle $\psi_t(n)$. The particles may also include additional attributes that are relevant to a particular application. In the case of FVTHMM, this includes $\mathcal{C}_t(n)$, an attribute denoting the counter vector. The task of the PBDT algorithm is then to systematically generate such particles at each time step or whenever new measurement arrives, while allowing for state inferences to be done efficiently.

The data structure of such particles can be visualised as a forest of trees with a link pointing from each child to its parent, like shown in Figure 4.3. Some particles may not have arrows leading back to them as they do not have children given their low scores relative to others. Throughout the operation of the algorithm, this structure will be maintained while the score for each $n$th particle is updated via the recursion

$$\mathcal{S}_t(n) = \begin{cases} \mathcal{B}_1(n), & \text{if } t = 1 \\ \mathcal{S}_{t-1}(m) + \mathcal{U}_t(n), & \text{if } t > 1, \end{cases} \tag{4.8}$$

where $\mathcal{B}_1(n)$ signifies the initial base score, $\mathcal{U}_t(n)$ is the stage cost used for updating the score and $m$ is the parent of $n$th particle at time $t$, i.e. $m = \psi_t(n)$.

Henceforth, we shall refer to $\mathcal{X}_t(n)$ as the system state estimate of the $n$th particle at time $t$ and $m$ as the index of parent of the same particle (i.e. $\mathcal{X}_t(n) = \hat{\mathbf{x}}_t^{(n)}$ and $m = \psi_t(n)$), while $\hat{\mathbf{x}}_{1:t}^{(n)}$ denotes the system state trajectory obtained from the concatenation of $\hat{\mathbf{x}}_t^{(n)}$, $\hat{\mathbf{x}}_{t-1}^{(m)}$, and so on until that of the parent of the corresponding particle at time $t = 2$, i.e. $\hat{\mathbf{x}}_{1:t}^{(n)} = (\ldots, \hat{\mathbf{x}}_{t-1}^{(m)}, \hat{\mathbf{x}}_t^{(n)})$. In this regard, each $n$th trajectory ending at time $t$, $\hat{\mathbf{x}}_{1:t}^{(n)}$, could be a candidate solution for explaining the observations $y_{1:t}$. The same relation applies to $\mathcal{C}_t(n)$, $\hat{\mathbf{c}}_t^{(n)}$ and $\hat{\mathbf{c}}_{1:t}^{(n)}$. For notational convenience, we will also group the fields of each $n$th particle into a tuple $\mathcal{P}_t(n) = (\mathcal{X}_t(n), \psi_t(n), \mathcal{C}_t(n), \mathcal{S}_t(n))$, whereas $\mathcal{P}_t$ refers to the list of generated particles at time $t$, $\overline{\mathcal{P}}_t$ is the version of $\mathcal{P}_t$ which is sorted in descending order of the score, and $\tilde{\mathcal{P}}_t$ is the truncated version of $\overline{\mathcal{P}}_t$, as we shall see later.

Under FVTHMM, and in line with the recursive expression of the joint probability in (3.7), the terms of (4.8) take the following more concrete form,

- $\mathcal{B}_1(n) := \log(p(\hat{\mathbf{x}}_1^{(n)})) + \log(p(y_t \mid \mathbf{x}_1^{(n)}))$
- $\mathcal{S}_t(n) := \log(p(\hat{\mathbf{x}}_{1:t}^{(n)}, y_{1:t}))$
- $\mathcal{U}_t(n) := \log(p(\hat{\mathbf{x}}_t^{(n)}, \hat{\mathbf{c}}_t^{(n)} \mid \hat{\mathbf{x}}_{t-1}^{(m)}, \hat{\mathbf{c}}_{t-1}^{(m)})) + \log(p(y_t \mid \hat{\mathbf{x}}_t^{(n)}))$,

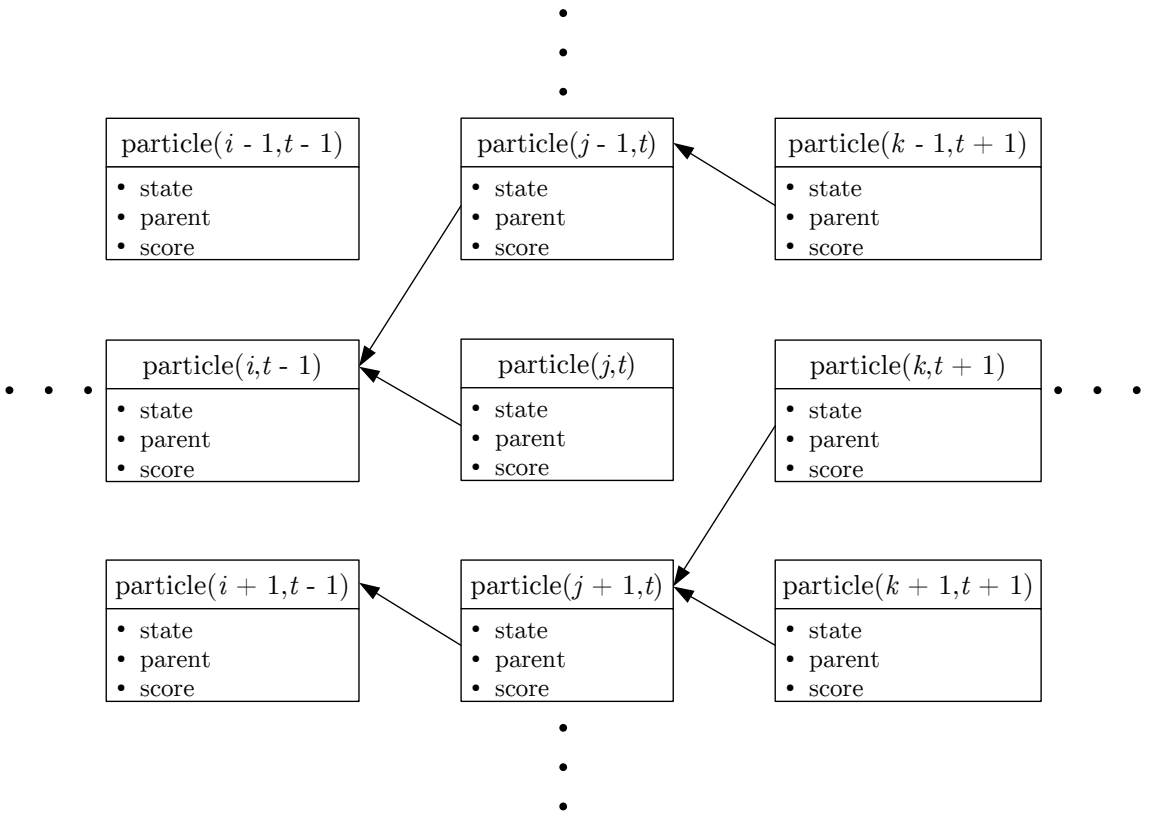



Figure 4.3: Data structure of particles.

resulting in

$$\mathcal{S}_t(n) = \begin{cases} \log(p(\hat{\mathbf{x}}_1^{(n)})) + \log(p(y_t \mid \mathbf{x}_1^{(n)})), & \text{if } t = 1 \\ \\ \mathcal{S}_{t-1}(m) + \log(p(y_t \mid \hat{\mathbf{x}}_t^{(n)})) \\ + \log(p(\hat{\mathbf{x}}_t^{(n)}, \hat{\mathbf{c}}_t^{(n)} \mid \hat{\mathbf{x}}_{t-1}^{(m)}, \hat{\mathbf{c}}_{t-1}^{(m)})) & , \quad \text{if } t > 1. \end{cases} \tag{4.9}$$

For the purpose of describing PBDT, suppose that we have already generated $N_{\text{p}}(t-1)$ particles at time $t-1$ (i.e. $\tilde{\mathcal{P}}_{t-1}$) and we would like to generate a new list of particles for the current time step $t$ (i.e. $\tilde{\mathcal{P}}_t$). The naive approach entails enumerating all possible next states for each parent particle, evaluating the score or likelihood of those states, before finally keeping a maximum of the $N_{\text{p,max}}$ highest-scoring particles. However, this is wasteful and computationally inefficient, since we can typically tell in advance the states which are not going to be among the $N_{\text{p,max}}$ kept particles. As such, it is normally the case that not all possible states need to be considered.

To that end, with consideration of the ways to enumerate a substantially reduced set of possible states, and with reference to Figure 4.4, the three distinct stages – (1) state-pruning, (2) combine and sort, (3) merge and truncate – involved in the generation of particles at each time step are presented in the discussion that follows.

### State-pruning

The first stage involves a state-pruning procedure, where three criteria are utilised for eliminating states that are unlikely or impossible. The first criterion being the consideration of only states that correspond to at most three simultaneous state transitions. The rationale is that it is unlikely to have more than three appliances switching state simultaneously at a given time step, an observation which is well reflected in Figure 4.5. Formally, this condition can be expressed as $d_H(\mathbf{x}_{t-1}, \mathbf{x}_t) \leq 3$ where $d_H(\mathbf{x}_{t-1}, \mathbf{x}_t)$ denotes the Hamming distance between $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$. Therefore, for the state of each $m$th parent particle, $\hat{\mathbf{x}}_{t-1}^{(m)}$, only values of $\mathbf{x}_t$ with $d_H(\hat{\mathbf{x}}_{t-1}^{(m)}, \mathbf{x}_t) \leq 3$ need to be considered in the enumeration.

On top of the that, we also exploit the sparsity in the emission probability factor to consider only possible states that satisfy the condition $p(y_t \mid \mathbf{x}_t) > \epsilon$, where $\epsilon$ is the machine precision of a microprocessor. Figure 4.6 shows an instance of this sparseness for one of the houses in the REDD dataset and highlights that the number of possible states is substantially lower than the number of states

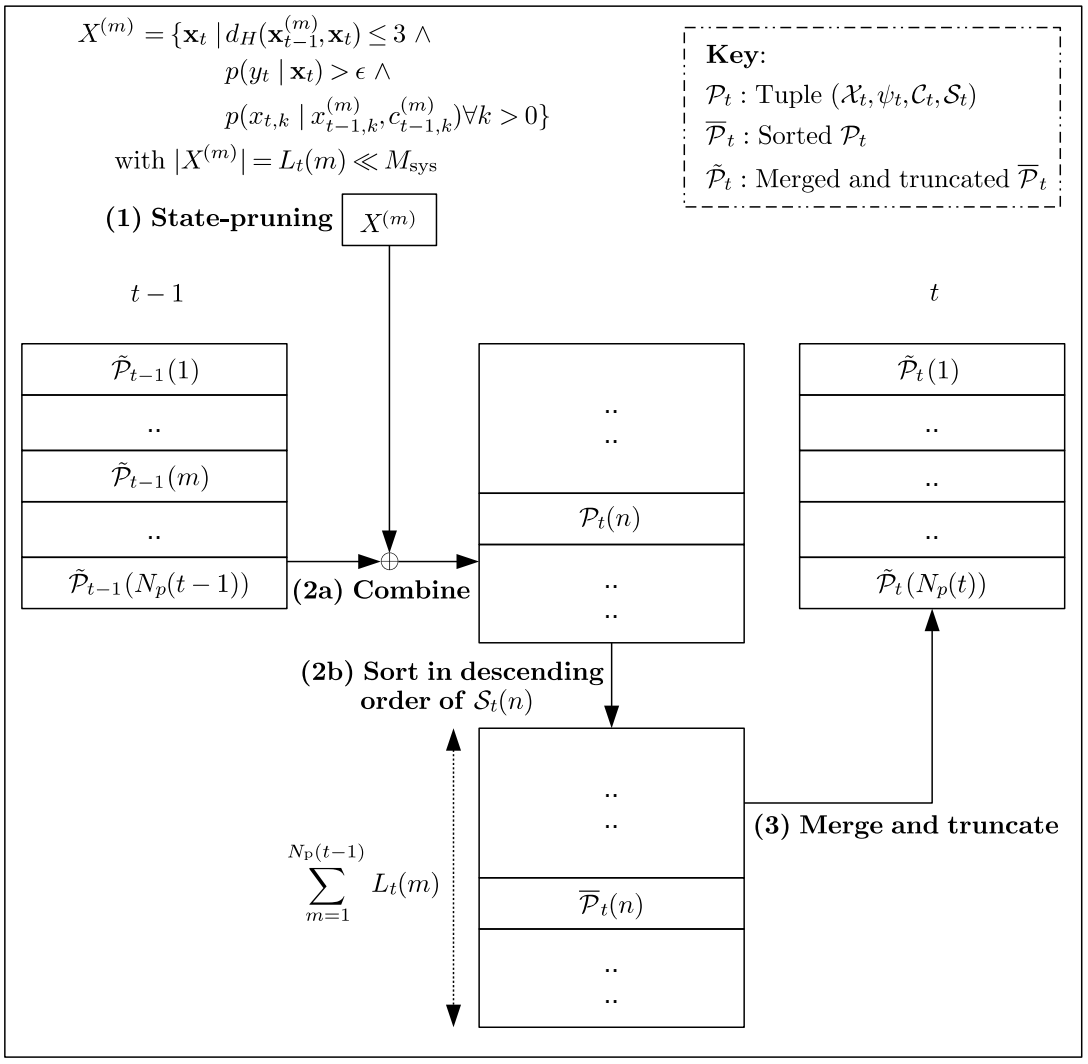



Figure 4.4: An overview of the particle generation procedure. The notations in the figure are explained in the main description of the algorithm.

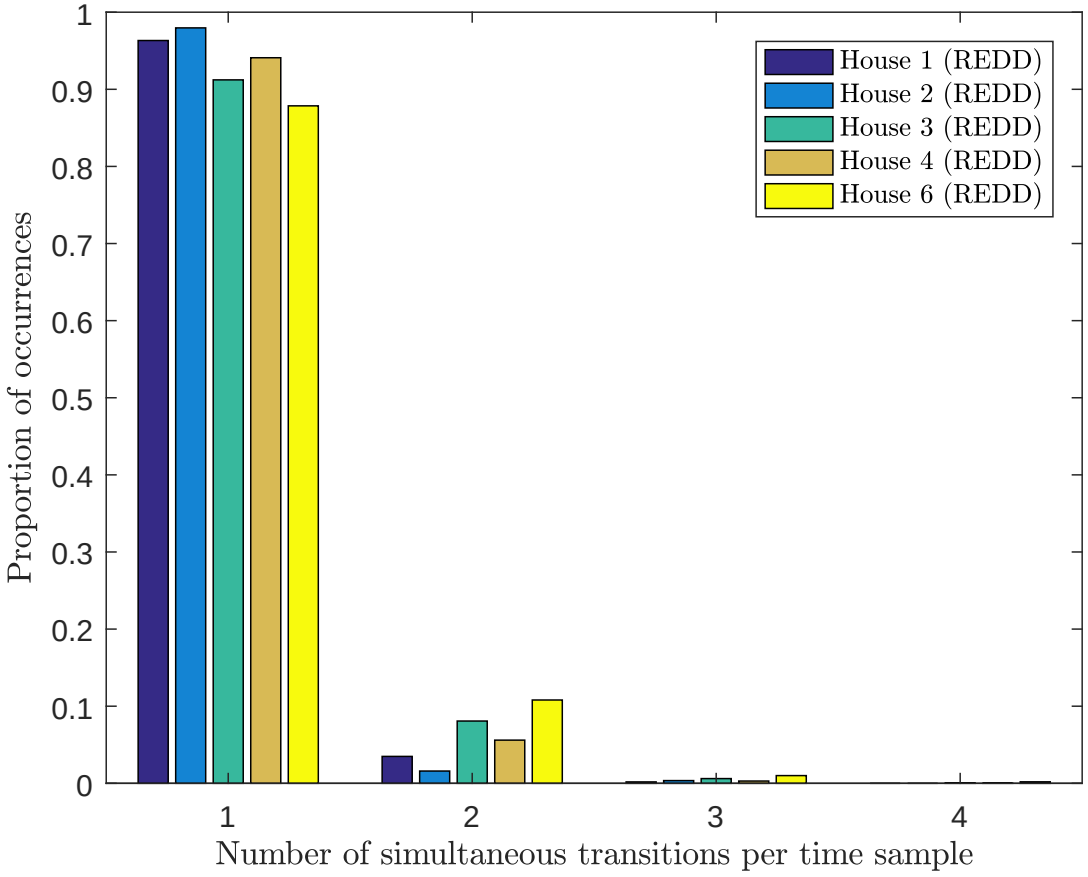



Figure 4.5: The proportion of occurrences for different number of simultaneous state transitions per time sample across the considered houses in the REDD dataset. The data used has been downsampled by a factor of 3 to result in a sampling interval of approximately 10 seconds.

inherent in the system, i.e. $M_{\text{sys}} = \prod_{k=1}^{K} M_k$, with $M_k$ being the number of states for appliance $k$. This property can be attributed to the relatively small number of states which are consistent with the constraint $y_t = \sum_{k=1}^{K} y_{t,k}$, an effect that can be seen in Figure 4.7. Taking advantage of this allows computational cost to be further reduced.

In a more general case however, the state-pruning need not be based on the sparsity of the emission factor. If other factors can be easily computed and their sparsity is high, they could be utilised for accelerating computation as well. One such example specific to FVTHMM forms the third pruning criterion. It takes advantage of the factorisation of $p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{c}_{t-1})$, where the knowledge of any single zero factor $p(x_{t,k} = j_k \mid \hat{x}^{(m)}_{t,k-1}, \hat{c}^{(m)}_{t-1,k})$ for a given $m$th particle enables any state vector $\mathbf{x}_t$ with the corresponding $k$th element being $j_k$ to be discarded. For example, if we know that the probability of transitioning to state $j_k$ of appliance $k$ is 0 given certain values of $(\hat{x}^{(m)}_{t,k-1}, \hat{c}^{(m)}_{t-1,k})$, then all system states $\mathbf{x}_t$ whose $k$th component is $j_k$ can be disregarded. This means, progressively, each realisation of a zero factor allows the state space over the possible $\mathbf{x}_t$ to be reduced by a factor of $\frac{M_k^*}{M_k^*-1}$, with $M_k^*$ being the current count of possible values of $x_{t,k}$.

By consolidating the aforementioned three criteria, the outcome of the state-pruning procedure is a reduced set of possible states which each $m$th particle at time $t-1$ could transition to, i.e.

$$X^{(m)} = \{\mathbf{x}_t \mid d_H(\mathbf{x}^{(m)}_{t-1}, \mathbf{x}_t) \leq 3 \wedge \\ p(y_t \mid \mathbf{x}_t) > \epsilon \wedge p(x_{t,k} \mid x^{(m)}_{t-1,k}, c^{(m)}_{t-1,k}) \forall k > 0\}.$$

The cardinality of $X^{(m)}$ is $L_t(m) = |X^{(m)}|$, and as the size of the reduced set is very much smaller than the cardinality of the full state space, i.e. $L_t(m) \ll M_{\text{sys}}$, significant computational savings can be realised. Note that this pruning does not require explicit testing of each of the $M_{\text{sys}}$ possible next states, but allows a fairly efficient enumeration of $X^{(m)}$. Overall, this concludes the first stage of the algorithm.

#### Combine and sort

In the second stage, the set of $L_t(m)$ potential state candidates obtained previously are combined with their respective parent particles to form offspring particles. The process starts by taking each state candidate $j$ and computing the score that would result from the transition of the state associated with the $m$th parent particle, $\hat{\mathbf{x}}^{(m)}_{t-1}$, to $j$. This is followed by updating the counter vector of each $n$th

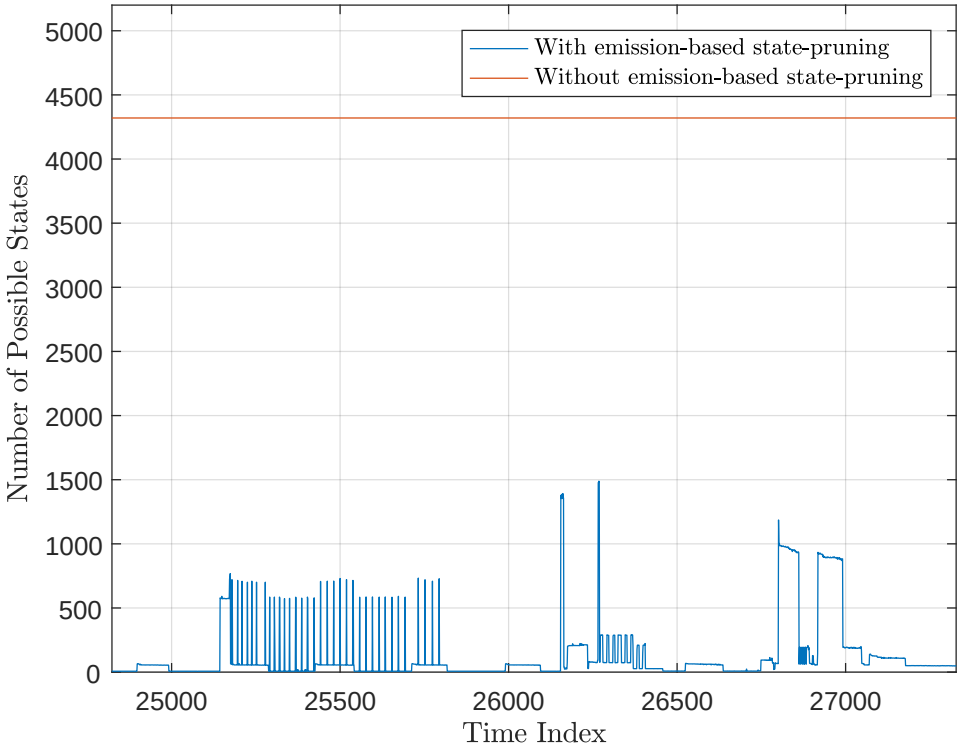



Figure 4.6: The number of possible states for different values of observed aggregate power consumption $y_t$ across time for a short segment from house 2 of the REDD dataset.

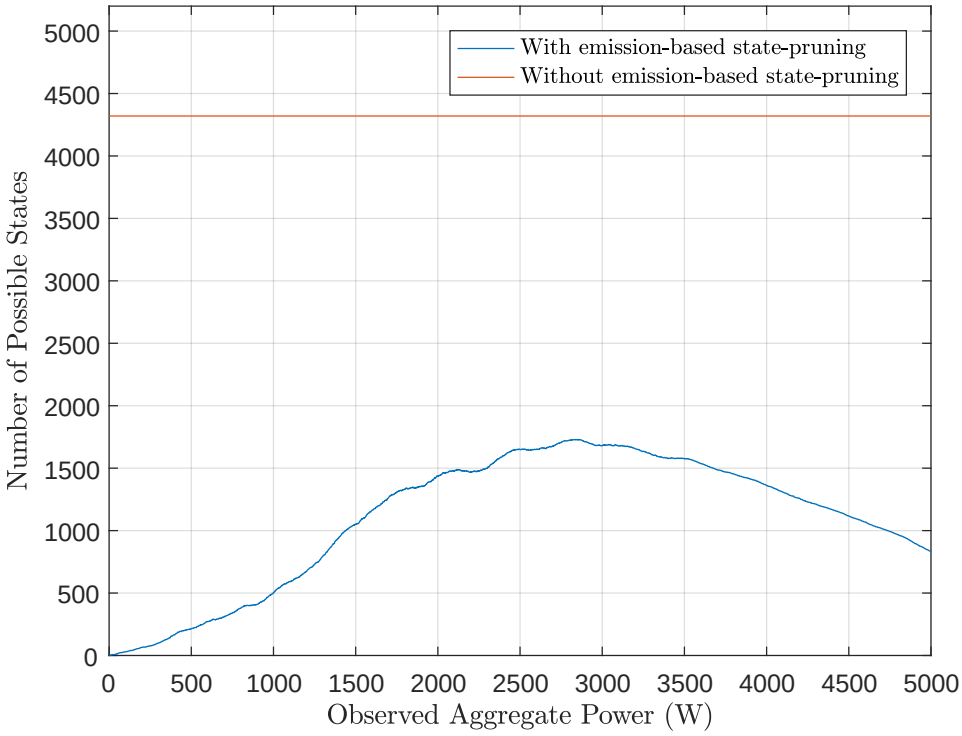



Figure 4.7: The effect of the observed aggregate power consumption on the number of possible states for house 2 of the REDD dataset. This is based on the emission probability $p(y_t \mid \mathbf{x}_t)$.

particle, $\hat{\mathbf{c}}_t^{(n)}$, according to (3.11). By repeating the calculation for each $j$ and each $m$, and by updating the reference to the parent particles, we obtain an intermediate list of particles $\mathcal{P}_t$ with length $\sum_{m=1}^{N_\mathrm{p}(t-1)} L_t(m)$, as shown in Figure 4.4. Using the computed score $\mathcal{S}_t$, the particles are then sorted to result in $\overline{\mathcal{P}}_t$.

### Merge and truncate

Finally, using a Viterbi-like operation in the third stage, the $\sum_{m=1}^{N_\mathrm{p}(t-1)} L_t(m)$ offspring particles are merged, keeping only the highest ranked particle for a given augmented system state $(\mathbf{x}_t, \mathbf{c}_t)$. This is made possible by the assumption in FVTHMM that the current system state is only dependent on the previous augmented system state (see Section 3.3.1 of Chapter 3). Therefore, much like the Viterbi algorithm, only the best particle amongst those with the same $(\mathbf{x}_t, \mathbf{c}_t)$ needs to be kept. In the implementation, the merging is done via a hash-based deduplication over the tuple $(\overline{\mathcal{X}}_t, \overline{\mathcal{C}}_t)$ using the MurmurHash3 algorithm [App16] and the result is a reduced set of $N_\mathrm{p}(t)'$ particles.

At this point, if $N_\mathrm{p}(t)'$ exceeds the user-defined maximum number of particles to keep at each time step, $N_{\mathrm{p,max}}$, the $N_{\mathrm{p,max}}$ particles with the highest score are retained. Otherwise, all of them are kept. The third stage marks the end of a single round of the particle generation procedure shown in Figure 4.4. With the arrival of a new power measurement at time $t+1$, the same procedure is repeated. A functional overview of the algorithm is listed in Algorithm 1.

### Backtracking

The method for generating new particles at each time step has been described. To determine the most likely sequence of states up until the current time, backtracking is used, as in the Viterbi algorithm. This works by taking the state of the best ranked particle at time $T$ (i.e. $\tilde{\mathcal{X}}_T(1)$) and the corresponding reference to the parent particle $\tilde{\psi}_T(1)$, then iteratively reading off the entry of the parent particle's state and its parent all the way back to $t = 1$ (see Line 26 to Line 31 of Algorithm 1). The result is $\hat{\mathbf{x}}_{1:T}$, the estimated state sequence as determined by PBDT for a given observation sequence $y_{1:T}$, a given set of model parameters $\boldsymbol{\lambda}$ and a given $N_{\mathrm{p,max}}$.

However, backtracking need not be performed at the end of a batch of data; it can be done on a continual basis. For example, in an actual set-up in a real-world setting where measurements are continuously being sampled in real-time, backtracking can be done on demand from a selected time slice, while the main

**Algorithm 1** Particle-Based Distribution Truncation

**function** PBDT($y_{1:T}, \boldsymbol{\lambda}, N_{\text{p,max}}$)
  **for** $t \leftarrow 1$ to $T$ **do**
    $n \leftarrow 1$
    **for** $m \leftarrow 1$ to $N_{\text{p}}(t-1)$ **do**
      $i \leftarrow \tilde{\mathcal{X}}_{t-1}(m)$
      $\tau \leftarrow \tilde{\mathcal{C}}_{t-1}(m)$
      $X = \{j \mid d_H(i,j) \leq 3 \wedge$
        $p(y_t \mid \mathbf{x}_t = j) > 0 \wedge$
        $p(x_{t,k} = j_k \mid x_{t-1,k} = i_k, c_{t-1,k} = \tau_k) \forall k > 0\}$
      **for all** $j \in X$ **do**
        Update $\mathcal{S}_t(n)$ using (4.9)
        $\psi_t(n) \leftarrow m$
        $\mathcal{X}_t(n) \leftarrow j$
        Update $\mathcal{C}_t(n)$ based on $\mathbf{c}_t$ using (3.11)
        $n \leftarrow n + 1$
      **end for**
    **end for**
    $\mathcal{P}_t \leftarrow (\mathcal{X}_t, \psi_t, \mathcal{C}_t, \mathcal{S}_t)$
    $\overline{\mathcal{P}}_t \leftarrow$ SORT($\mathcal{P}_t, \mathcal{S}_t$,'Desc') ▷ Sort $\mathcal{P}_t$ in descending order of $\mathcal{S}_t$
    $\tilde{\mathcal{P}}_t \leftarrow$ DEDUPLICATE($\overline{\mathcal{P}}_t, (\overline{\mathcal{X}}_t, \overline{\mathcal{C}}_t)$) ▷ Deduplicate $\overline{\mathcal{P}}_t$ over $(\overline{\mathcal{X}}_t, \overline{\mathcal{C}}_t)$
    **if** LENGTH($\tilde{\mathcal{P}}_t$) $> N_{\text{p,max}}$ **then**
      $\tilde{\mathcal{P}}_t \leftarrow \tilde{\mathcal{P}}_t(1\!:\!N_{\text{p,max}})$
      $N_{\text{p}}(t) \leftarrow N_{\text{p,max}}$
    **else**
      $N_{\text{p}}(t) \leftarrow$ LENGTH($\tilde{\mathcal{P}}_t$)
    **end if**
  **end for**
  $\hat{\mathbf{x}}_T \leftarrow \tilde{\mathcal{X}}_T(1)$
  $\hat{\psi}_T \leftarrow \tilde{\psi}_T(1)$
  **for** $t \leftarrow T-1$ to $1$ **do**
    $\hat{\mathbf{x}}_t \leftarrow \tilde{\mathcal{X}}_t(\hat{\psi}_{t+1})$
    $\hat{\psi}_t \leftarrow \tilde{\psi}_t(\hat{\psi}_{t+1})$
  **end for**
  **return** $\hat{\mathbf{x}}_{1:T}$
**end function**

particle generation procedure continues to process new samples as they come. To provide some context in such a setting, we can consider an in-home display unit (IHD) with a button that could be pressed by the user to switch from the display of the aggregate power consumption to the estimated appliance-level power consumption as resulting from the tentatively most likely particle.

Although results from such preliminary backtracking are liable to change as new data comes in, states sufficiently far in the past are not susceptible to this change. We call the point from which change is guaranteed not to occur as the fusion point. If we let $\Psi_t = \{m \mid m = \tilde{\psi}_t(n) \forall n \in \{1, \ldots, N_{\mathrm{p}}(t)\}\}$ and define for some $\tau < t$

$$\Psi_\tau = \{m \mid m = \tilde{\psi}_\tau(n),\ \exists n \in \Psi_{\tau+1}\},$$

then backtracking can be done automatically from the fusion point $(t^*, n^*)$ such that $t^* \in \{\tau^* \mid |\Psi_{\tau^*+1}| = 1 \wedge \tau^* < t\}$ and $n^* \in \Psi_{t^*+1}$. An example of such a fusion point is shown in Figure 4.8, where it is noted that all particles at $t = 6000$ have the first-rank particle (i.e. particle with the highest score) at $t = 5608$ as a common ancestor. As it turns out, the concept of fusion point has been similarly explored for the case of real-time Viterbi decoding in previous work [BR08].

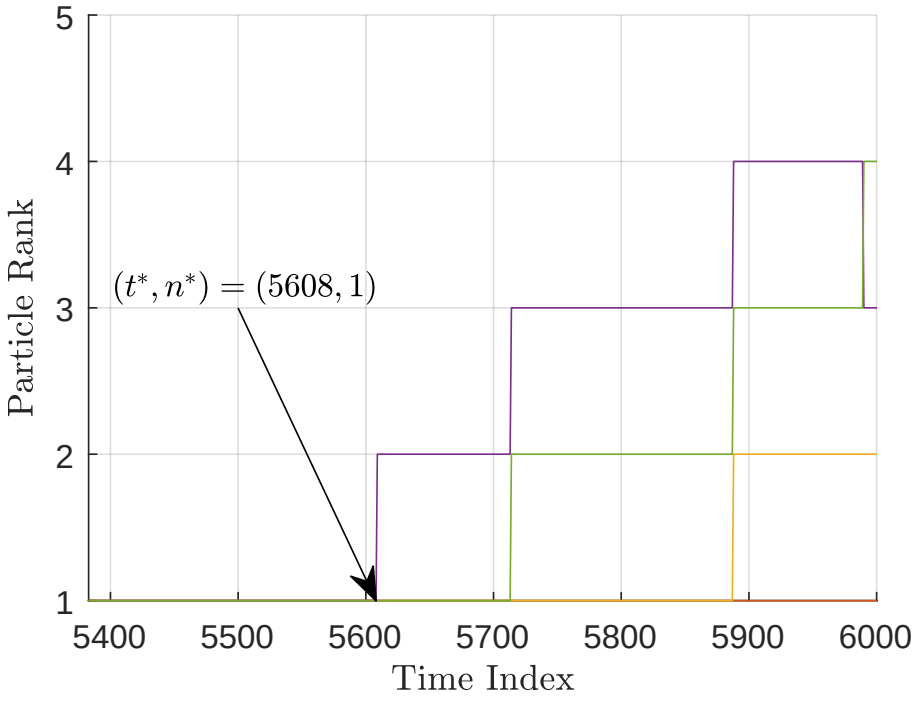



Figure 4.8: Every particle at $t = 6000$ has the first-rank particle at $t = 5608$ as a common ancestor. Therefore, the fusion point is as indicated.

Overall, the correctness of such a backtracking approach is validated by the observation that no further measurements beyond $t$ could change the fusion point (i.e. new measurements do not change the common ancestor). Hence, $\hat{\mathbf{x}}_{1:t^*}^{(n^*)}$ can be taken as the final estimated state trajectory from the beginning of time to $t^*$. This allows the memory utilised by particles of these times to be reused for storing future particles, a process that is in some ways similar to the concept of garbage collection in the computer science literature [WJNB95]. Additionally, reclaiming memory is such a way is beneficial for the PBDT algorithm, considering that a naive implementation of PBDT requires memory that is $\Omega(N_{\mathrm{p,max}}T)$, i.e. the required memory space grows at least proportionally with $N_{\mathrm{p,max}}T$.

In the next subsection, we will discuss a few other important aspects of the PBDT algorithm that should be considered from an implementation perspective.

### 4.3.2 Implementation Remarks

For the purpose of this research, the PBDT algorithm has been implemented as a MATLAB function, with multiple subroutines written in C/C++ using the MEX[1] API, for speeding up computations.

While the algorithm can be implemented as it is listed in Algorithm 1, we have chosen to incorporate a number of additional computational optimisations at the implementation level, following a few observations noted during the course of the algorithm's development. Exploiting these observations allows the enumerations of the reduced set of possible states, the evaluations of the per-appliance hazard function $h_{x_{t-1,k}}(\cdot)$ and the computations of the emission probabilities $p(y_t \mid \mathbf{x})$, all of which are needed as part of the score calculation in Line 9 of Algorithm 1, to be shared across particles which are similar in some fundamental sense. This eliminates redundant computations, enabling further speed improvements to be realised.

#### Sharing the computation results of the hazard function

Recall that, in generating the particles for a new time step $t$ given a list of particles in the previous time step $t-1$ (i.e. parent particles), the algorithm has to compute the score for $L_t(m)$ new particles for each $m$th parent particle. With $N_{\mathrm{p}}(t-1)$ parent particles, there would be $\sum_{m=1}^{N_{\mathrm{p}}(t-1)} L_t(m)$ such computations, and in each

[1] MEX is a library and a set of application programming interfaces (API) provided by MATLAB for calling compiled code written in C/C++. It is typically used for speeding up time-critical operations.

computation, the hazard function has to be evaluated once for each appliance $k$ so that the duration-dependent state transition probability $p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{c}_{t-1})$ could be calculated. This means, there are $K \cdot \sum_{m=1}^{N_\text{p}(t-1)} L_t(m)$ evaluations of the hazard function at each time step. However, because the hazard function $h_{x_{t-1,k}}(c_{t-1,k})$ is only dependent on both the state and dwell time of appliance $k$, the computation of $h_{x_{t-1,k}}(c_{t-1,k})$ can be shared across parent particles with the same augmented device state $(x_{t-1,k}, c_{t-1,k})$, despite the uniqueness in the augmented *total* system state $(\mathbf{x}_{t-1}, \mathbf{c}_{t-1})$ among all the parent particles (by virtue of Line 18 in Algorithm 1).

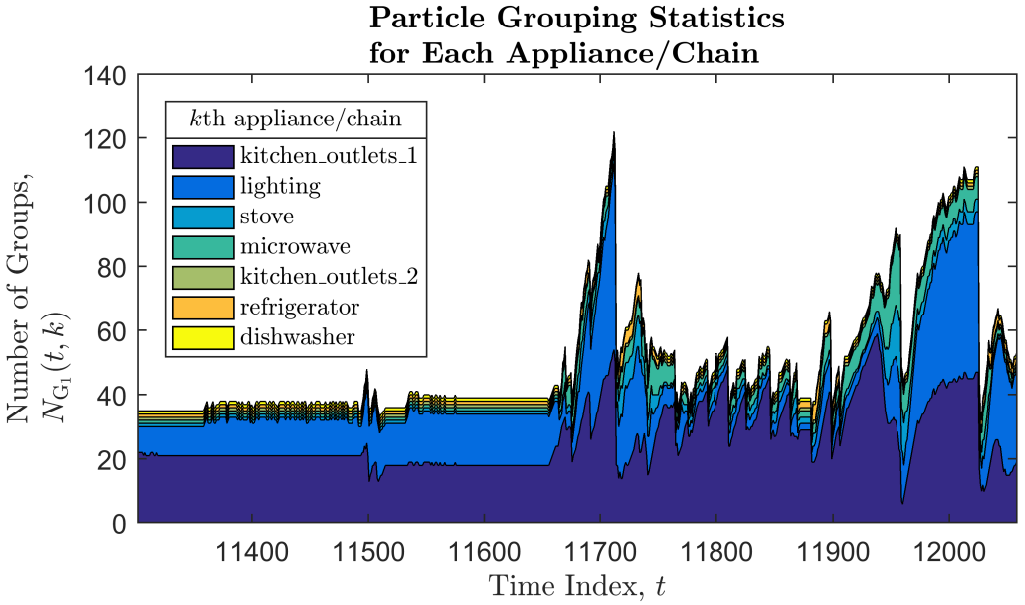
In this way, groups of similar parent particles could be formed, allowing the calculation of the hazard function $h_{x_{t-1,k}}(\cdot)$ to be done only once for each group of parent particles, and enabling the results of the calculation to be reused by parent particles of the same group. If $N^{(g)}_{\text{p},\text{G}_1}(t,k)$ is the number of such particles in group $g$ and $N_{\text{G}_1}(t,k)$ is the number of different groups for a particular time $t$ and device $k$, then we have the constraints

$$1 \leq N_{\text{G}_1}(t,k) \leq N_\text{p}(t), \quad \forall t \text{ and } \forall k \tag{4.10}$$

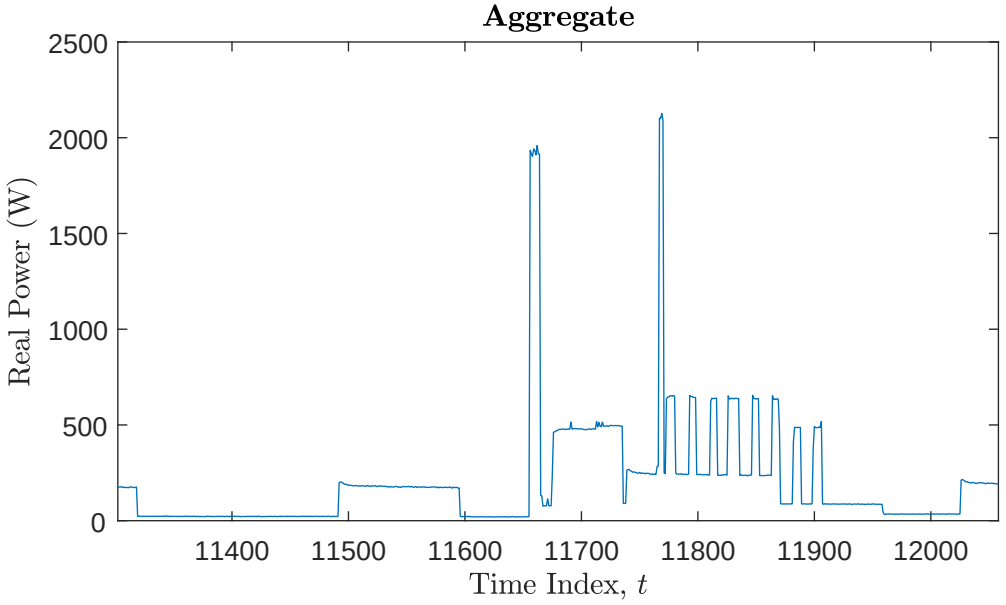
$$\sum_{g=1}^{N_{\text{G}_1}(t,k)} N^{(g)}_{\text{p},\text{G}_1}(t,k) = N_\text{p}(t), \quad \forall t \text{ and } \forall k. \tag{4.11}$$

The upper bound of the first constraint is met when $(x_{t,k}, c_{t,k})$ is unique across all particles (i.e. there are as many groups as the number of particles at time $t$) while the lower limit of 1 occurs when $(x_{t,k}, c_{t,k})$ is the same for all particles (i.e. there is only one group). The second constraint is merely a natural fact that the total number of particles across each group has to sum to $N_\text{p}(t)$. It is easily seen that, when $N_{\text{G}_1}(t,k) = 1$, the highest speed-up occurs. Conversely, no speed-up should be expected when $N_{\text{G}_1}(t,k) = N_\text{p}(t)$.

During the course of the algorithm's development, situations with $N_{\text{G}_1}(t,k) = N_\text{p}(t)$ were found to be rare and the grouping of particles is beneficial in reducing the number of computations. As an illustrative example, consider Figure 4.9a, where it is shown that the number of groups $N_{\text{G}_1}(t,k)$ for different appliances is always less than $N_\text{p}(t)$, or in this case, $N_{\text{p,max}}$ of 100. In fact, besides **`kitchen_outlets1`** and **`lighting`**, all the other appliances usually have $N_{\text{G}_1}(t,k)$ of 1. This means, for the hazard function calculation for each of these appliances, only a single computation needs to be done in most cases.




(a) The height of each segment of the area plot corresponds to the number of distinct groups for a given chain/appliance.




(b) The segment of aggregate data that the PBDT algoritm is applied to. The data is from house 2 of the REDD dataset.

Figure 4.9: The number of distinct groups of particles with the same $(x_{t,k}, c_{t,k})$ at each time step.

Also illustrated in Figure 4.9 is how the observed aggregate power affects the outcome of the grouping. Such variation may be attributed to the existence of a dominant set of possible $(x_{t,k}, c_{t,k})$ over the observed $y_{1:t}$. For cases with many possible competing $(x_{t,k}, c_{t,k})$ that are more or less equally dominant, the number of groups will be large, and as a result, the number of particles within each group will be small. Interestingly, this can be seen as a certainty measure of $(x_{t,k}, c_{t,k})$; if majority of the particles generated at a certain time step $t$ have a common $(x_{t,k}, c_{t,k})$, then there is a high confidence associated with the augmented state estimates of appliance $k$. Otherwise, $(x_{t,k}, c_{t,k})$ among the particles are distributed more widely (i.e. large $N_{\mathrm{G}_1}(t,k)$) and there is a lower confidence in each of the estimates. Viewing in this way, it seems reasonable to devote more computational resources when there are more competing estimates. The observation that the sharing of computation results allows such dynamic usage of resources is appealing and it is an important part of the implementation of PBDT.

In the actual implementation considered for this research, grouping is performed before the start of the particle generation procedure at each time step. More specifically, each $m$th particle at $t-1$ is assigned to a group, by means of

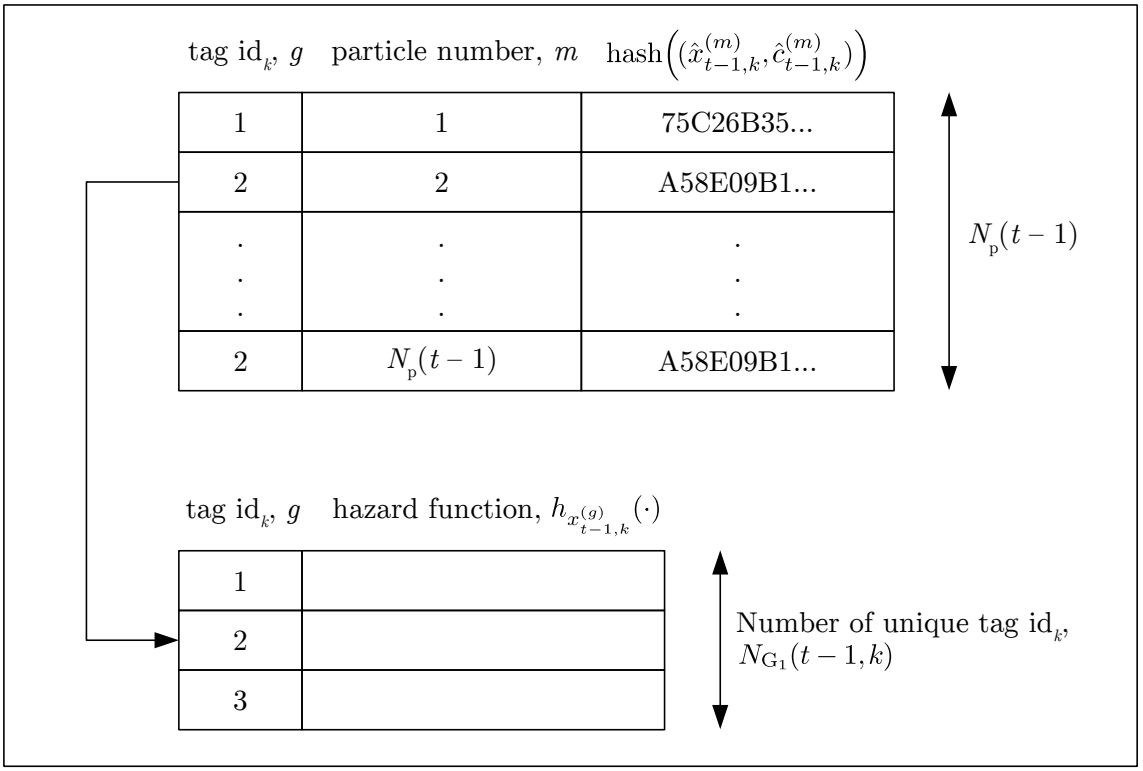



Figure 4.10: The grouping of parent particles and the precomputation of the hazard function values for each appliance $k$ before the start of the particle generation procedure at each time step.

hashing the augmented device state of appliance $k$, $(\hat{x}^{(m)}_{t-1,k}, \hat{c}^{(m)}_{t-1,k})$, and tagging each particle with an identification number depending on which group it belongs to (see Figure 4.10). Then, the hazard function corresponding to each group is precomputed and its result is stored into a secondary data structure to facilitate fast lookup during the particle generation procedure. Compared to the unoptimised implementation, which entails evaluating the hazard function $K \cdot \sum_{m=1}^{N_{\mathrm{p}}(t-1)} L_t(m)$ times, only $\sum_{k=1}^{K} N_{\mathrm{G}_1}(t-1,k)$ evaluations are needed with the sharing of computation results.

### Sharing the enumeration of possible states and the emission probability calculations

In very much the same way as the sharing of computational results of the hazard function discussed previously, the enumeration of possible states, $X^{(m)}$, for each $m$th parent particle, and the corresponding emission probability calculations, $p(y_t \mid \mathbf{x}_t \in X^{(m)})$, can be shared across parent particles with the same $\mathbf{x}_{t-1}$. This is because as described in Section 4.3.1, the outcome of the enumeration for a given time step is only affected by the system state of the previous time step, $\mathbf{x}_{t-1}$.

As such, parent particles with the same $\mathbf{x}_{t-1}$ can be grouped. For time $t$, each group $g$ has $N^{(g)}_{\mathrm{p},\mathrm{G}_2}(t)$ particles and the total number of groups is $N_{\mathrm{G}_2}(t)$, which altogether satisfy the constraints

$$1 \leq N_{\mathrm{G}_2}(t) \leq N_{\mathrm{p}}(t) \quad \forall t \tag{4.12}$$

$$\sum_{g=1}^{N_{\mathrm{G}_2}(t)} N_{\mathrm{p},\mathrm{G}_2}(t) = N_{\mathrm{p}}(t) \quad \forall t. \tag{4.13}$$

Like before, the outcome of the grouping is affected by the observed aggregate power and it determines the speed-up that could be obtained. A $N_{\mathrm{G}_2}(t)$ of 1 results in a highest speed-up, whereas no speed-up should be expected when $N_{\mathrm{G}_2}(t) = N_{\mathrm{p}}(t)$.

In the implementation, a data structure shown in Figure 4.11 is maintained to keep track of the group assignments, while a secondary data structure stores the reduced set of possible $\mathbf{x}_t$, $X^{(g)}$, as arising from each group $g$ of parent particles, and the corresponding $\log(p(y_t \mid \mathbf{x}_t))$. Before the start of the particle generation procedure at each time step, both data structures are updated with new precom-

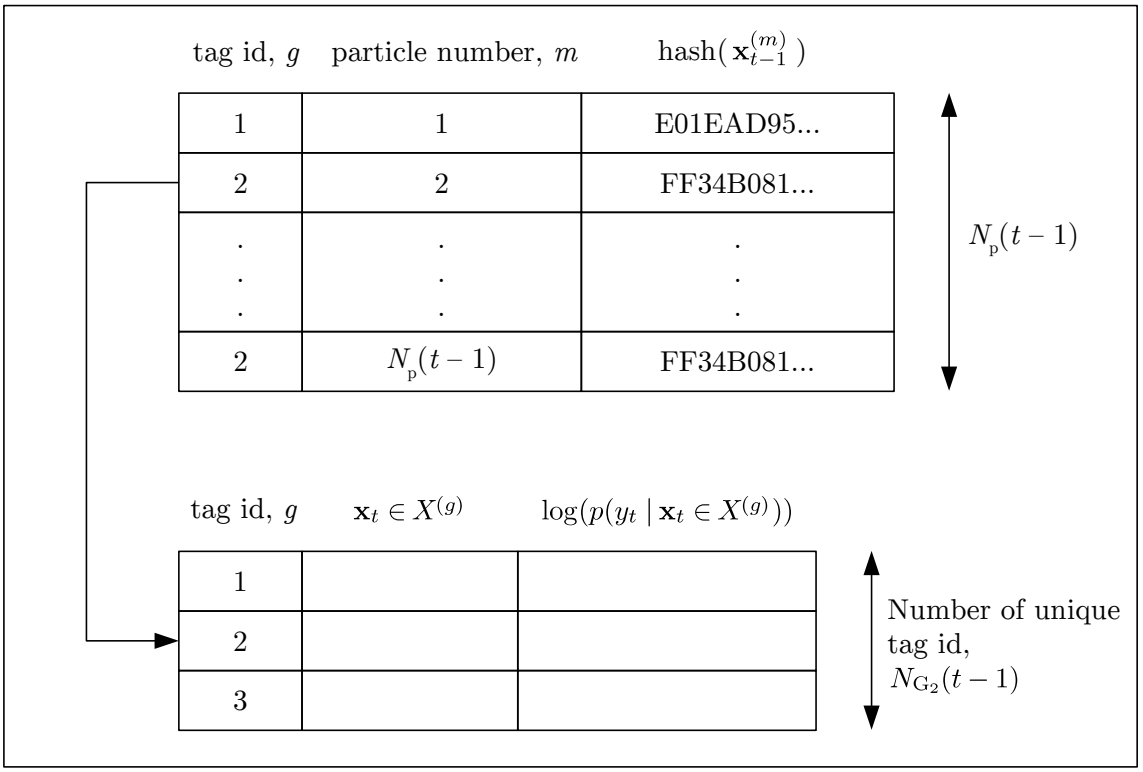



Figure 4.11: The precomputation of $\log(p(y_t \mid \mathbf{x}_t))$ for each enumerated set of possible states common to group $g$.

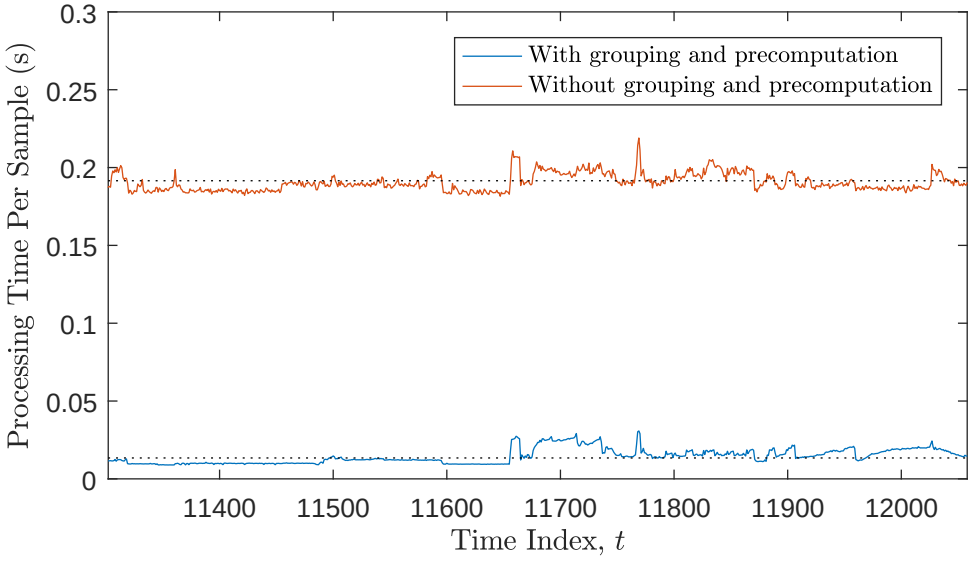



Figure 4.12: Comparison between the use of the precomputation scheme and without, in terms of the time taken to process each sample shown in Figure 4.9b.

puted values so that they can be looked up quickly later when new particles are actually generated.

#### Speed improvements

Although there are computational overheads in grouping the particles, the adoption of the aforementioned computational sharing schemes outweighs any such overheads and offers great improvements in computational efficiency. For example, when applying the PBDT algorithm to the data shown in Figure 4.9b, the time taken to process each sample is on average 20 times lower than when no grouping and precomputation were used. This comparison is illustrated in Figure 4.12.

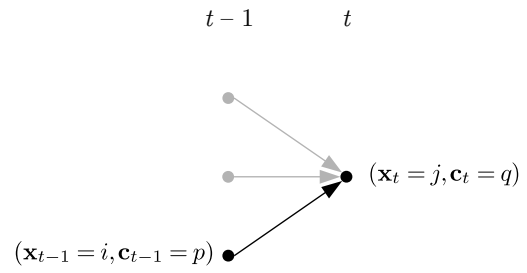
### 4.3.3 Relationship to the Viterbi Algorithm

At the start of Section 4.3, we have remarked that the PBDT algorithm is an approximation to the Viterbi algorithm. Here, we show how this arises naturally by considering a few conceptual similarities and differences between the former and the latter. Then, we discuss the influence of $N_{\text{p,max}}$ in controlling the extent of the approximation.

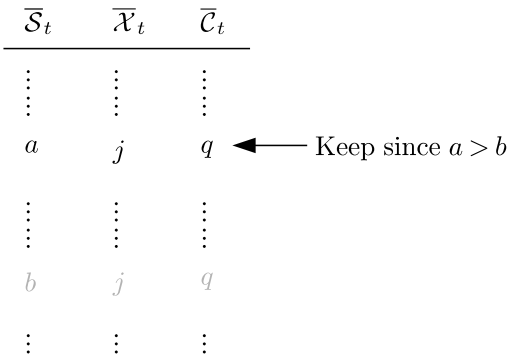
#### The merging step

The central part of the Viterbi algorithm is solving the Bellman equation [Bel03] to find the most likely previous state for each possible current state. Given that information, all other possible paths to each current state can be discarded. In PBDT, this exact operation is performed in the merging step. To understand this, consider one part of the trellis structure shown in Figure 4.13a. For any given destination augmented state $(j, q)$, there will be multiple originating states $(i, p)$. When presented with such a situation, the Viterbi algorithm records the selected $(i, p)$, owing to the Markov property over the transition from $(\mathbf{x}_{t-1}, \mathbf{c}_{t-1})$ to $(\mathbf{x}_t, \mathbf{c}_t)$. On the other hand, in the PBDT algorithm, each such transitions to $(j, q)$ is represented by a particle. Thus, to exploit the Markov property in this case, only the best particle amongst those with the same $(\mathbf{x}_t, \mathbf{c}_t)$ is kept (see Figure 4.13b). It is not difficult see that both the PBDT algorithm and the Viterbi algorithm are equivalent in this regard.

In all, the Viterbi version may seem simpler, considering that no explicit merging is involved. However, the simple operation hinges on the existence of a trellis structure. As this incurs a huge memory cost under FVTHMM, its creation is infeasible in a practical setting. Moreover, the allocation of such a structure is




(a) Selection of the best previous state for a given destination state.




(b) As shown is the pre-merge sorted list of particles. Among the particles with the augmented state vector of $(j, q)$, only the one with the highest score is kept.

Figure 4.13: Comparison between the PBDT algorithm and the Viterbi algorithm in terms of the merging process.

wasteful due to the indirect constraints imposed by (3.11) on possible values of $\mathbf{c}_t$. This, coupled with the fact that the counter values are only known during runtime means the preallocation of the trellis structure as required by the Viterbi algorithm is not only difficult but uneconomical, memory-wise.

### Log likelihood of the estimated state sequence

By definition, the sequence of states estimated using the Viterbi algorithm is optimal under a given model since it corresponds to the solution of the maximum likelihood problem. Given that the PBDT algorithm is an approximation method for solving the same maximum likelihood problem, it is expected that the log likelihood of the state sequence from the Viterbi algorithm, $\mathcal{L}_T^{\text{VT}}$, forms an upper bound to the log likelihood of the state sequence estimated using PBDT, $\mathcal{L}_T^{\text{PBDT}}$. However, if the number of particles to keep at each time time, $N_{\text{p,max}}$, is sufficiently large, we would expect the $\mathcal{L}_T^{\text{PBDT}}$ to converge to $\mathcal{L}_T^{\text{VT}}$. This gives us the following conjecture.

*Conjecture 4.1:* Let $\mathcal{L}_T^{\text{PBDT}}$ be the log likelihood of the estimated state sequence $\hat{\mathbf{x}}_{1:T}^{\text{PBDT}}$ from the PBDT algorithm and let $\mathcal{L}_T^{\text{VT}}$ be the log likelihood of the estimated state sequence $\hat{\mathbf{x}}_{1:T}^{\text{VT}}$ from the Viterbi algorithm. Then, $\mathcal{L}_T^{\text{PBDT}} \leq \mathcal{L}_T^{\text{VT}}$, with decreasing $\left|\mathcal{L}_T^{\text{VT}} - \mathcal{L}_T^{\text{PBDT}}\right|$ as $N_{\text{p,max}}$ is increased.

### The role of $N_{\mathbf{p,max}}$

Being the only parameter, $N_{\text{p,max}}$ controls the extent of the approximation inherent to the PBDT algorithm. Using a small $N_{\text{p,max}}$ has the potential to truncate away optimal states that would otherwise be considered by the Viterbi algorithm. Therefore, $N_{\text{p,max}}$ plays a major role in determining $\mathcal{L}_T^{\text{PBDT}}$ and the associated $\hat{\mathbf{x}}_{1:T}^{\text{PBDT}}$.

To investigate the influence of $N_{\text{p,max}}$ in practice, we apply both the Viterbi algorithm for FHMM and the PBDT algorithm for FHMM to a segment of test data from house 2 of the REDD dataset. The comparison using FVTHMM is not considered since the Viterbi algorithm is not computationally tractable under such circumstances, as noted in Section 4.2. In the test, the PBDT algorithm is run from $N_{\text{p,max}} = 1$ to $N_{\text{p,max}} = 100$, with incremental steps of 1. For each $N_{\text{p,max}}$, the corresponding $\mathcal{L}_T^{\text{PBDT}}$ is recorded. Also noted is the number of differences between $\hat{\mathbf{x}}_{1:T}^{\text{PBDT}}$ and $\hat{\mathbf{x}}_{1:T}^{\text{VT}}$ for each round.

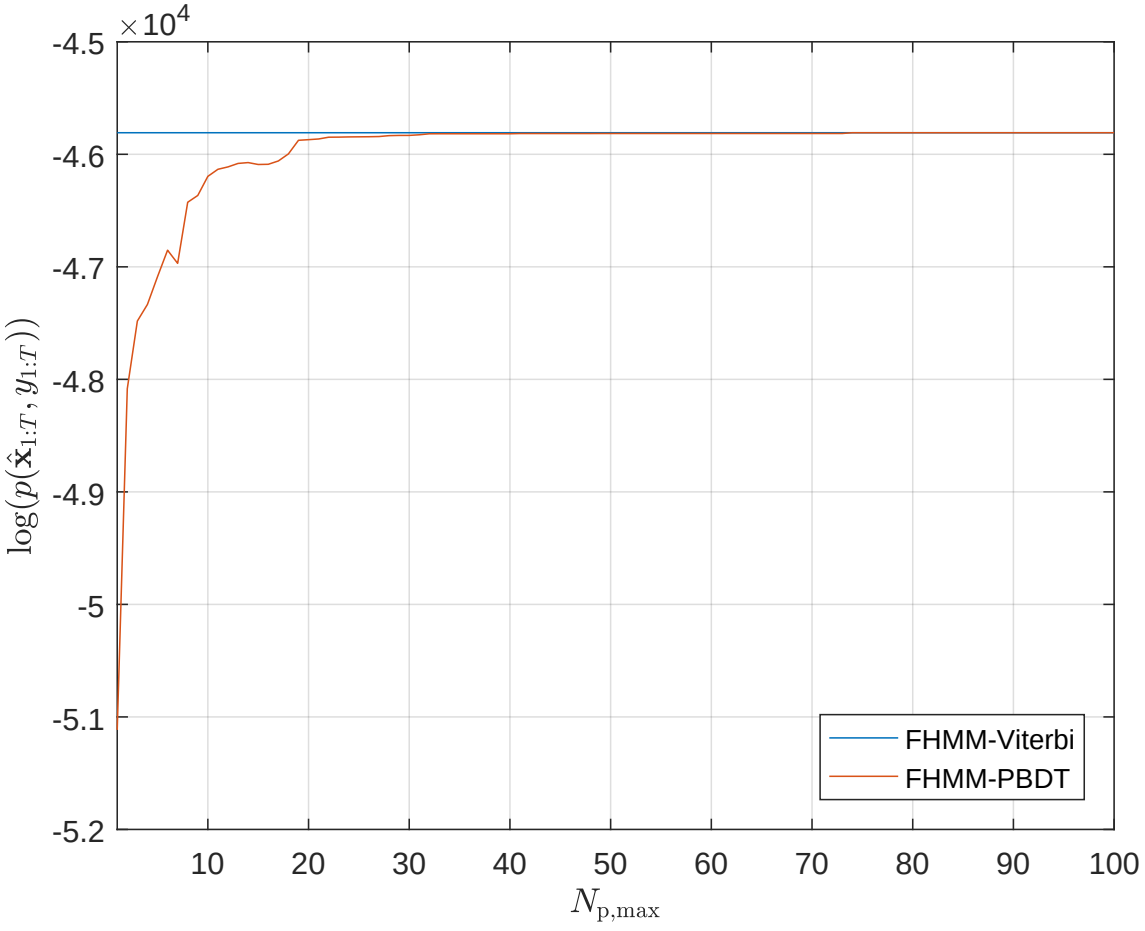



Figure 4.14: Effects of the $N_{\text{p,max}}$ parameter on the log likelihood of the estimated sequence.

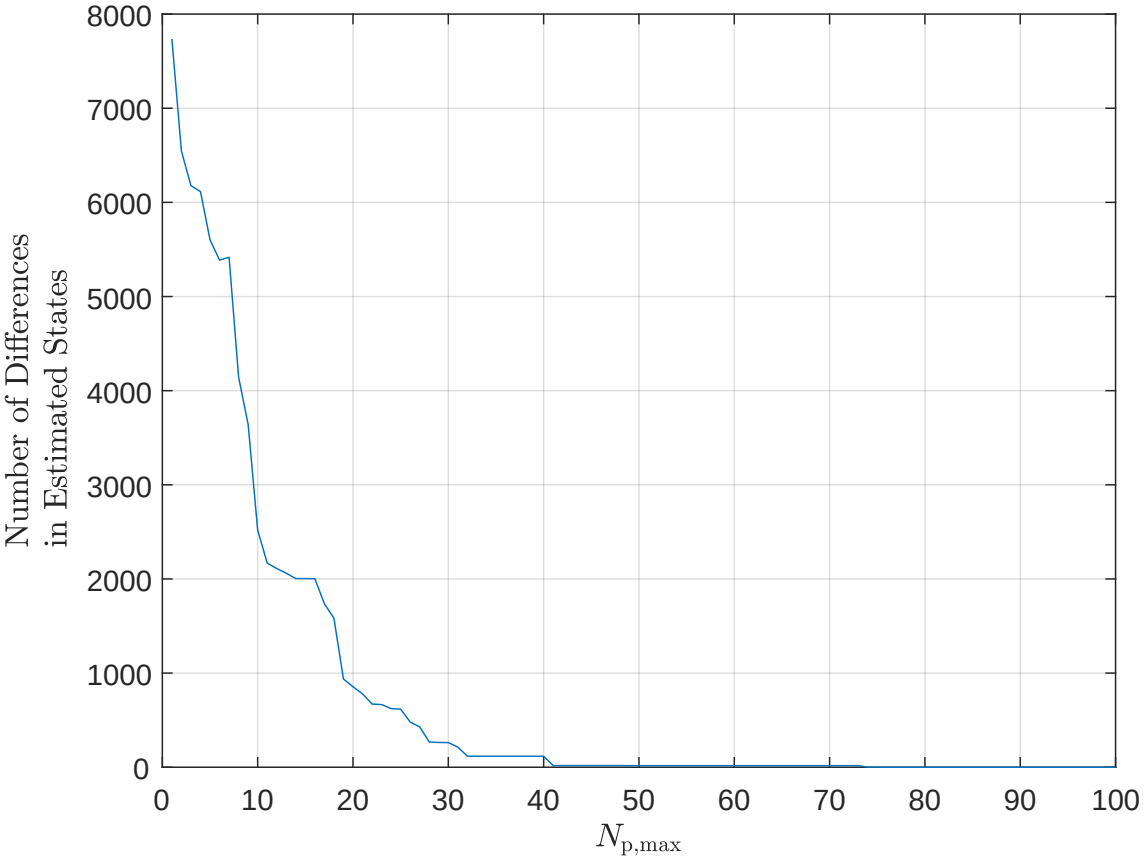



Figure 4.15: Effects of the $N_{\text{p,max}}$ parameter on the differences in estimates between the Viterbi algorithm and the PBDT algorithm.

The results of the test are presented in Figure 4.14 and Figure 4.15 respectively. As can be seen, $\mathcal{L}_T^{\text{PBDT}}$ converges to $\mathcal{L}_T^{\text{VT}}$ as $N_{\text{p,max}}$ increases, while the number of differences between $\hat{\mathbf{x}}_{1:T}^{\text{PBDT}}$ and $\hat{\mathbf{x}}_{1:T}^{\text{VT}}$ converges to zero. This is consistent with Conjecture 4.1 and the results not only validate the approximation nature of the PBDT algorithm but also the algorithm's natural convergence to the optimal solution[2], given a large enough $N_{\text{p,max}}$. At least for the test data considered, the equality between $\mathcal{L}_T^{\text{PBDT}}$ and $\mathcal{L}_T^{\text{VT}}$ happens at $N_{\text{p,max}} = 74$, a value that is much lower than the number of states intrinsic to house 2 of the REDD dataset (i.e. $M_{\text{sys}} = 4320$). However, this is not particularly surprising if we consider the state-pruning criteria imposed by the PBDT algorithm in reducing the number of effective states.

Apart from the results and its aforementioned roles, one other interesting aspect of $N_{\text{p,max}}$ is its relation to the greediness of the algorithm. When $N_{\text{p,max}} = 1$, the PBDT algoritm reduces to a greedy algorithm as only the locally-optimal solution at each time step is kept. In contrast, increasing $N_{\text{p,max}}$ away from 1 has

[2]A related later work by Lange and Bergés [LB16] (published in the late 2016 as alluded to at the end of Section 4.1) also illustrated how their algorithm improves as more HMM paths (analogous to $N_{\text{p,max}}$ in our case) are kept.

the opposite effect. This culminates to the PBDT algorithm being a full dynamic programming approach at the convergent point. Taken together, $N_{\text{p,max}}$ can be interpreted as an inverse-greediness parameter and it can be chosen depending on the available computational resources at hand.

## 4.4 Evaluation of Disaggregation Accuracy on Real-world Data

A NILM algorithm is often judged by how well it is able to reconstruct the appliance-level power consumption signal from the observed aggregate-level measurements. In this section, we will evaluate the ability of our approach in this regard by testing against a real-world dataset. We will also compare our algorithm with other benchmark approaches to gauge its disaggregation performance. All evaluations are done using MATLAB on a PC with an Intel Core i7-4770 processor and 16 GB of RAM, while a graphical software application was developed to facilitate the understanding of the estimates given by the PBDT algorithm (see Appendix 2). However, before going into the experimental configurations and the evaluations, we will first detail a few metrics that are pertinent to disaggregation accuracies and the classification of errors.

### 4.4.1 Evaluation Metrics

For quantifying disaggregation accuracy, we will adopt the Correct Assignment Rate (CAR) metric introduced by [KJ11] and used by [KDM+16, KDH+16, MPB+16, JW13]. The metric can be defined as

$$\text{CAR} = 1 - \frac{\sum_{t=1}^{T}\sum_{k=1}^{K}|\hat{y}_{t,k} - y_{t,k}|}{2\sum_{t=1}^{T} y_t}, \tag{4.14}$$

where $\hat{y}_{t,k}$ and $y_{t,k}$ are the estimated power consumption and the actual power consumption of the $k$th appliance at time $t$ respectively.

In addition, we will also introduce a new metric based on the energy associated with the true positives ($\text{E}_{\text{TP}}$), the false negatives ($\text{E}_{\text{FN}}$) and the false positives ($\text{E}_{\text{FP}}$). Each of these has units of kilowatt-hour (kWh). For the $k$th appliance, they are given as

$$\text{E}_{\text{TP},k} = \int_0^T \min(\hat{y}_{t,k}, y_{t,k})dt \tag{4.15}$$

$$\mathrm{E}_{\mathrm{FN},k} = \mathrm{E}_k^* - \mathrm{E}_{\mathrm{TP},k} \tag{4.16}$$

$$\mathrm{E}_{\mathrm{FP},k} = \hat{\mathrm{E}}_k - \mathrm{E}_{\mathrm{TP},k}, \tag{4.17}$$

where $\mathrm{E}_k^*$ denotes the actual energy consumed by the $k$th appliance and $\hat{\mathrm{E}}_k$ refers to the estimated energy consumed by the $k$th appliance.

### 4.4.2 Classification of Errors

Apart from the emphasis on disaggregation accuracy, it is also important to consider potential causes of errors in state estimations. This is especially beneficial in the case of the PBDT algorithm as errors can be a result of approximations due to truncations or due to simplifications in the model itself.

However, before introducing a new metric to quantify the nature of errors, let us first introduce a few definitions. If we denote the true state vector at time $\tau$ and the estimated state vector at time $\tau$ to be $\mathbf{x}_\tau^*$ and $\hat{\mathbf{x}}_\tau$ respectively, then an error is an event when $\hat{\mathbf{x}}_\tau \neq \mathbf{x}_\tau^*$. By extension, an error segment can be defined as a sequence of consecutive errors delimited by times with no errors. This is illustrated in Figure 4.16 where $\hat{\mathbf{x}}_{t_1:t_2} \neq \mathbf{x}_{t_1:t_2}^*$.

Now, to better understand the errors, we will introduce a new metric, the Cumulative Error Log Likehood Ratio (CELLR). For the error segment shown in

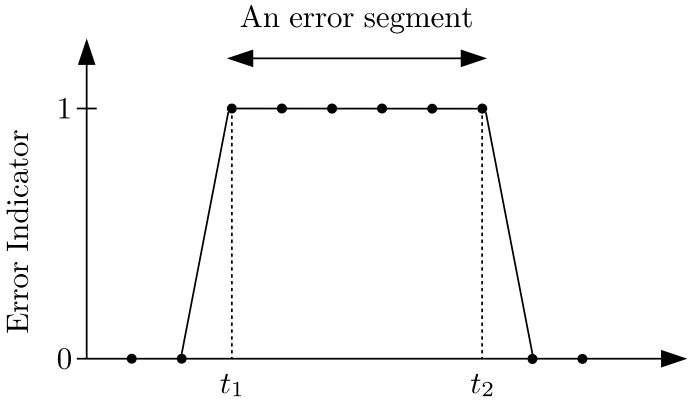



Figure 4.16: An error segment.

Figure 4.16, the metric can be defined as

$$\begin{aligned}\text{CELLR} &= \log\left(\frac{p(\hat{\mathbf{x}}_{t_1:t_2+1}, y_{t_1:t_2+1} \mid \hat{\mathbf{x}}_{1:t_1-1}, \hat{\mathbf{c}}_{1:t_1-1}, y_{1:t_1-1})}{p(\mathbf{x}^*_{t_1:t_2+1}, y_{t_1:t_2+1} \mid \mathbf{x}^*_{1:t_1-1}, \mathbf{c}^*_{1:t_1-1}, y_{1:t_1-1})}\right) \\ &= \sum_{\tau=t_1}^{t_2+1} \log\left(\frac{p(\hat{\mathbf{x}}_\tau \mid \hat{\mathbf{x}}_{\tau-1}, \hat{\mathbf{c}}_{\tau-1}) p(y_\tau \mid \hat{\mathbf{x}}_\tau)}{p(\mathbf{x}^*_\tau \mid \mathbf{x}^*_{\tau-1}, \mathbf{c}^*_{\tau-1}) p(y_\tau \mid \mathbf{x}^*_\tau)}\right),\end{aligned} \tag{4.18}$$

where the denominator in the log function signifies the probability of assigning the true state vectors over an error segment given the assignments prior to the onset of the error, while the corresponding numerator denotes that of the estimates from the algorithm. A positive CELLR for a given error segment implies model-induced errors, whereas a negative CELLR suggests that the errors are not due to the model. In this context, model-induced errors could refer to errors owing to modelling assumptions such as the assumption of a Gaussian distribution for the data or other assumptions that do not reflect reality.

The basis of this error classification rule leverages two facts. Firstly, the estimated state sequence resulting from $\arg\max_{\mathbf{x}_{1:T}, \mathbf{c}_{1:T}} p(\mathbf{x}_{1:T}, y_{1:T}, \mathbf{c}_{1:T})$ is by definition the optimal state sequence under the model used. If the optimal state sequence is different from the actual state sequence and the actual state sequence has a lower log likelihood value than that of the optimal state sequence, it is deemed that the model used for the optimisation does not sufficiently capture the behaviour of the dynamical system under consideration.

Secondly, by definition, the log likelihood of the optimal state sequence forms an upper bound to the log likelihood of the state sequence estimated using the PBDT algorithm. In the event that the log likelihood of the actual state sequence is still lower than that of the potentially suboptimal state sequence from the PBDT algorithm, then the conclusion that follows from the first fact continues to hold. On the other hand, for the inverse case, we can say with high confidence that the model is not the primary cause for errors. In fact, they may very well be attributed to the suboptimal optimisation as part of the PBDT algorithm, owing perhaps to the use of the lower-than-required $N_{\text{p,max}}$. Taken together, it is this interplay between the aforementioned two facts that the basis for the CELLR metric and the associated error classification rule is founded upon.

For quantifying model-induced errors at a deeper level, let us also consider two additional metrics based on the decomposition of CELLR, such that

$$\text{CELLR} = \text{CELLR}_\text{e} + \text{CELLR}_\text{d}, \tag{4.19}$$

where $\text{CELLR}_\text{e}$ and $\text{CELLR}_\text{d}$ are the Cumulative Error Log Likelihood Ratio for the emission model and the state duration model, respectively. For the hypothetical error segment shown in Figure 4.16, they are each given as

$$\text{CELLR}_\text{e} = \sum_{\tau=t_1}^{t_2+1} \log\left(\frac{p(y_\tau \mid \hat{\mathbf{x}}_\tau)}{p(y_\tau \mid \mathbf{x}^*_\tau)}\right) \tag{4.20}$$

$$\text{CELLR}_\text{d} = \sum_{\tau=t_1}^{t_2+1} \log\left(\frac{p(\hat{\mathbf{x}}_\tau \mid \hat{\mathbf{x}}_{\tau-1}, \hat{\mathbf{c}}_{\tau-1})}{p(\mathbf{x}^*_\tau \mid \mathbf{x}^*_{\tau-1}, \mathbf{c}^*_{\tau-1})}\right). \tag{4.21}$$

Both $\text{CELLR}_\text{e}$ and $\text{CELLR}_\text{d}$ consider the log likelihood value of the actual state sequence (within an error segment) that would have been assigned by the emission model and the state duration model alone. Therefore, in a similar vein to the error classification rule of the CELLR metric, we can say that a positive $\text{CELLR}_\text{e}$ and a negative $\text{CELLR}_\text{d}$ signify that a given error segment is likely to be caused by the emission model as opposed to the state duration model or vice versa. However, unlike the case for CELLR, caution should be taken when making attributions because in performing state estimation, both the log likelihood for the emission model and the state duration model contribute to the overall likelihood. Unless it is clear that one is more dominant than the other, it is difficult to fully ascertain the cause by using these two metrics alone. Nevertheless, the use of all three metrics – CELLR, $\text{CELLR}_\text{e}$ and $\text{CELLR}_\text{d}$ – together would provide a greater insight on the nature of errors that might occur in the disaggregation of the real-world aggregate-level data. Thus, they will be used in part for explaining the results in the sections that follow.

### 4.4.3 REDD Dataset

To evaluate how well the PBDT algorithm with FVTHMM (FVTHMM-PBDT) performs in the context of real-world data, we have chosen to use the publicly available REDD dataset [KJ11]. The primary reasons are, the REDD dataset is widely regarded by the NILM community as the de facto dataset for benchmarking NILM algorithms (used by [JW13, PGWR12, SLS14, Zei12, KDM+16, ES15, EBE15], among others), and a common reference dataset facilitates the comparison of different methods, whether those developed in the past or those that will be developed as part of any future work.

As a whole, the dataset consists of power data collected from 6 houses in the Greater Boston area for a period of up to nearly two months. For each house, the

aggregate power consumption data and the appliance-level power measurements are metered at time granularity of seconds. This is in addition to the provided aggregate-level high-frequency voltage and current signals, each sampled at a rate of 15kHz. However, we will not consider the latter as our work primarily focuses on the disaggregation of data representative of those that can be obtained from smart meters.

### 4.4.4 Experimental Configuration

In the evaluation, house 1, 2, 3, 4 and 6 in the REDD dataset are all considered for validating the robustness and scalability of our approach. House 5 is notably not used, given that it contains too many days of missing data (see Figure 4.17). Incidentally, this view is also shared by [FRA13] and [ES15] among others.

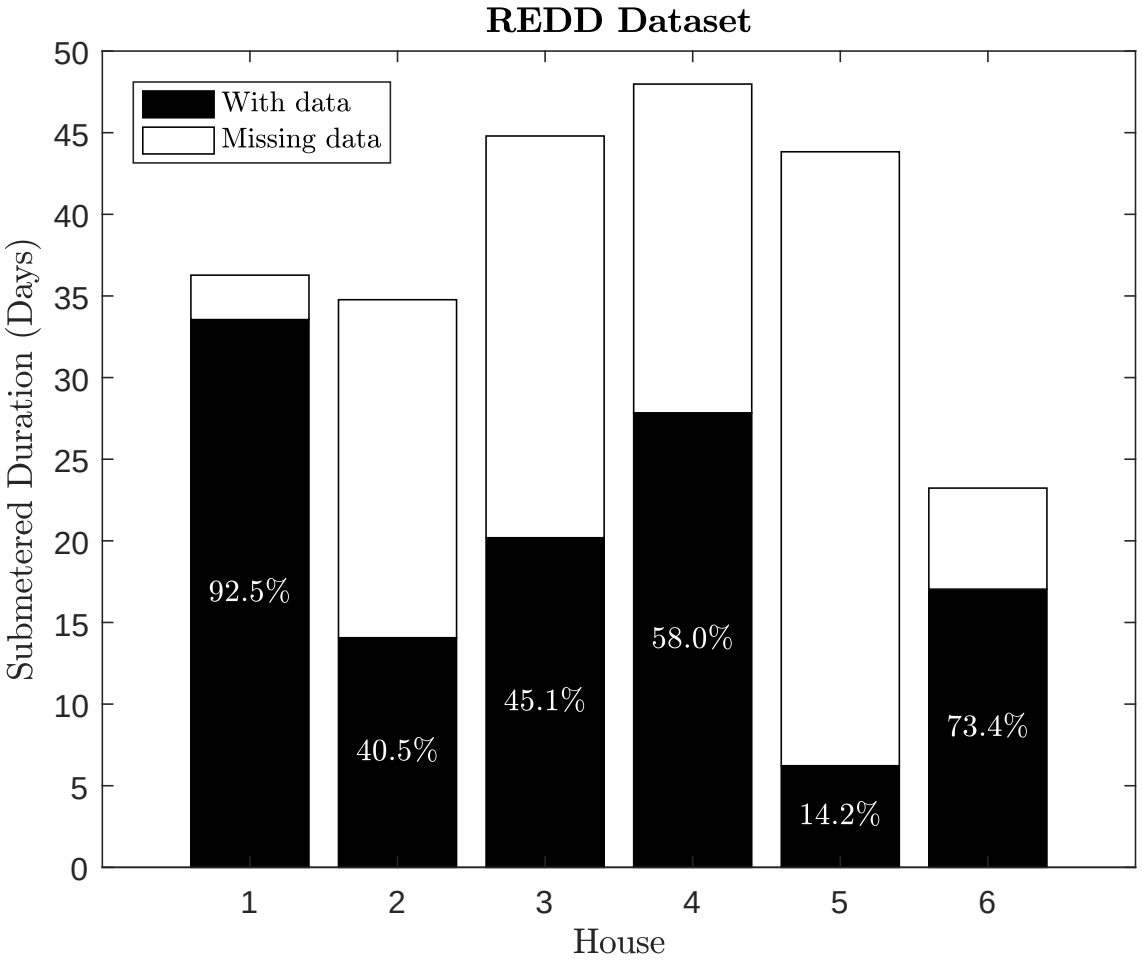



Figure 4.17: The duration of submetered data for each house in the REDD dataset. Times with no data are likely due to data transmission loss or the collection hardware being turned off inadvertently.

For each of the houses considered, there exist some appliances that were not turned ON predominantly during the monitoring period. As these appliances do not have sufficient ON-state data to be included in the analysis, they are removed from consideration. A summary of the appliances in question is presented in Table 4.1.

The aggregate data used for testing is constructed by summing up all the contributions of the involved submeter measurements. Then, they are downsampled by a factor of 3 by discarding every other sample. The result is data with a sampling interval of approximately 9 seconds, up from an original of 3. In the test, the aggregated data taken from the time interval shown in Figure 4.18 is treated as the test set, while data outside this range is used as part of the training stage for building appliance models. The missing data in the dataset is disregarded from disaggregation.

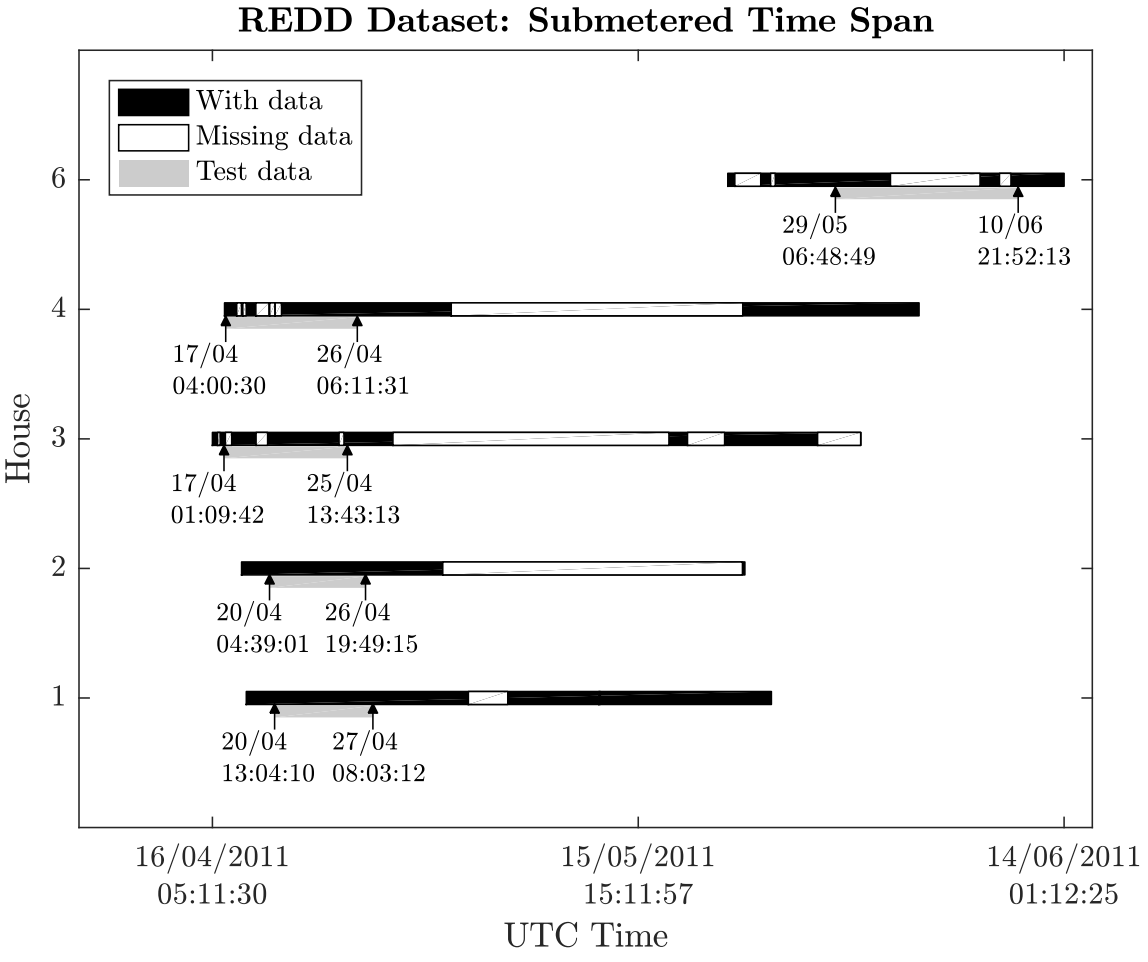



Figure 4.18: A detailed overview of the submetered time span of the REDD dataset. Sections with data, sections with missing data and sections used for testing are as indicated.

Table 4.1: Appliances in the REDD dataset.

| Submeter Label | House | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 6 |
| air_conditioning1 | | | | ■ | ■ |
| air_conditioning2 | | | | ■ | ■ |
| air_conditioning3 | | | | ■ | ■ |
| bathroom_gfi | ■ | | ■ | | ■ |
| bathroom_gfi1 | | | | ■ | |
| bathroom_gfi2 | | | | ■ | |
| dishwasher | ■ | ■ | ■ | ■ | ■ |
| disposal | | ■ | ■ | | |
| electric_heat | ■ | | | | ■ |
| electronics | | | ■ | | ■ |
| furnace | | | ■ | ■ | |
| kitchen_outlets1 | ■ | ■ | ■ | ■ | ■ |
| kitchen_outlets2 | ■ | ■ | ■ | ■ | ■ |
| kitchen_outlets3 | ■ | | | | |
| kitchen_outlets4 | ■ | | | | |
| lighting | | ■ | | | ■ |
| lighting1 | ■ | | ■ | ■ | |
| lighting2 | ■ | | ■ | ■ | |
| lighting3 | ■ | | ■ | ■ | |
| lighting4 | | | ■ | ■ | |
| lighting5 | | | ■ | | |
| microwave | ■ | ■ | ■ | | |
| miscellaneous | | | | ■ | |
| outlets_unknown | | | | ■ | |
| outlets_unknown1 | | | ■ | | ■ |
| outlets_unknown2 | | | ■ | | ■ |
| outlets_unknown3 | | | ■ | | |
| oven1 | ■ | | | | |
| oven2 | ■ | | | | |
| refrigerator | ■ | ■ | ■ | | ■ |
| smoke_alarms | | | ■ | ■ | |
| stove | ■ | ■ | | ■ | ■ |
| washer_dryer | | ■ | | ■ | ■ |
| washer_dryer1 | ■ | | ■ | | |
| washer_dryer2 | ■ | | ■ | | |
| washer_dryer3 | ■ | | | | |

**Key**:

■ In house # and considered

■ In house # but never turned on

□ Not in house #

Note that the aggregate-level measurements made available in the REDD dataset is not used for testing. This is driven by the less-documented issue of the data not actually being in real power quantities but rather in apparent power [BDS13, Zei12]. This mismatch in units between the aggregate data and the submeter data necessitates an ad-hoc preprocessing phase to convert the submeter data into apparent power quantities before training can be done to obtain the appliance models [BDS13]. In our view, such an act for dealing with the mismatch is not exactly accurate as power factor is required for the conversion between real power and apparent power, but it is not made available. With that, and to have a consistent validation of the estimated real power against the unprocessed submeter ground truth in the dataset, we have thus chosen to disregard the aggregate apparent power and instead, derive the test aggregate real power data as essentially the sum of the all the submeter data. Admittedly, this would impose the assumption that none of the test aggregate real power is contributed by unmetered appliances. However, we note that this is not an impediment to the evaluation of the methods considered. The core objective of disaggregation is still being tested but with the assumption that models of all appliances could be obtained. In Chapter 5, we take note of any deviation from the latter and propose a more robust extension to the techniques developed in this chapter.

### 4.4.5 Algorithm Configuration

The algorithms that will be considered in the comparison are

- PBDT with FVTHMM (FVTHMM-PBDT)
- PBDT with FHMM (FHMM-PBDT)
- Particle Filter with FVTHMM (FVTHMM-PF)
- Particle Filter with FHMM (FHMM-PF), a recent work by Egarter et al. [EBE15].

The Viterbi algorithm with FHMM (FHMM-Viterbi) will only be tested on house 2 of the REDD dataset since the number of states involved is only computationally tractable for this case. For all the other houses, the use of FHMM-Viterbi is intractable and the performance of FHMM-Viterbi can be taken to be close to FHMM-PBDT by virtue of Conjecture 4.1. The same can be said for FVTHMM-Viterbi in all cases. In the experiments, all particle-based algorithms will be configured to use a $N_{\mathrm{p,max}}$ of 100.

**PF-specific configuration**

Recall from Section 2.6.2 that an important aspect of particle filters is in the selection of the proposal distribution to sample from. In the comparison, we will use

$$q(\mathbf{x}_{1:t}) = p(\mathbf{x}_1) \prod_{\tau=1}^{t} p(\mathbf{x}_\tau \mid \mathbf{x}_{1:\tau-1}, y_\tau) \tag{4.22}$$

with samples being drawn incrementally from $p(\mathbf{x}_t \mid \mathbf{x}_{1:t-1}, y_t)$ at each time step. This proposal distribution has a corresponding weight function of

$$w(\mathbf{x}_{1:t}) = \prod_{\tau=1}^{t} p(y_\tau \mid \mathbf{x}_{1:\tau-1}) \tag{4.23}$$

and it will be used for resampling when the Effective Sample Size (ESS) criterion [DJ08] drops below 50.

### 4.4.6 Results and Discussion

Figure 4.19 shows a comparison between the methods considered in the experiment. Recall that, due to computational intractability, FHMM-Viterbi is only applied to house 2 of the REDD dataset. This is reflected in Figure 4.19 where bar graphs denoting the CAR metric of FHMM-Viterbi are omitted for all houses except for house 2. The same can be said for Table 4.2.

Table 4.2: CAR of different methods when applied to the REDD dataset.

| **CAR Metric** | | | | | | |
|---|---|---|---|---|---|---|
| **Methods** | **House** | | | | | **Average** |
| | 1 | 2 | 3 | 4 | 6 | |
| FVTHMM-PBDT | 76.96% | 82.87% | 80.59% | 64.47% | 78.63% | 76.70% |
| FHMM-PBDT | 58.51% | 67.34% | 78.73% | 46.24% | 62.53% | 62.71% |
| FVTHMM-PF | 69.26% | 84.00% | 62.72% | 62.14% | 69.32% | 69.49% |
| FHMM-PF [EBE15] | 54.22% | 82.73% | 61.96% | 57.10% | 68.23% | 64.85% |
| FHMM-Viterbi | N/A | 67.34% | N/A | N/A | N/A | N/A |

Overall, our proposed method (i.e. FVTHMM-PBDT) performs consistently well, recording the highest average CAR (i.e. 77%) among others. The biggest gain comes from the comparison against FHMM-PF in house 1, achieving a 22% improvement in disaggregation accuracy. Visual differences of the disaggregation results for a day's worth of data from house 1 is shown in Figure 4.20.

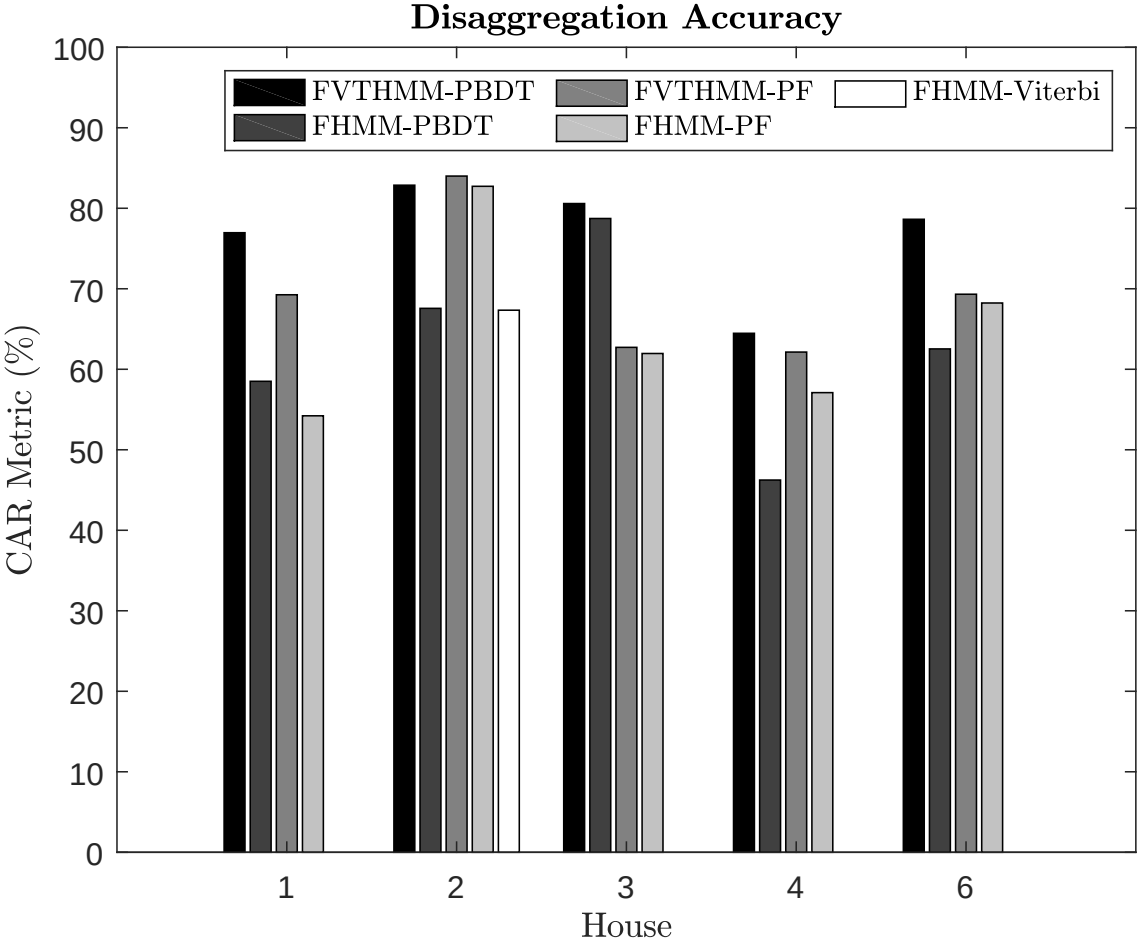
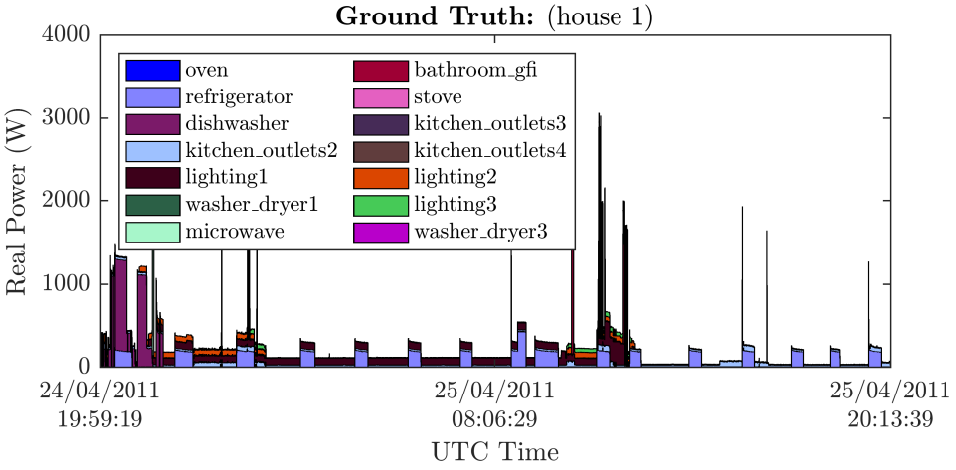
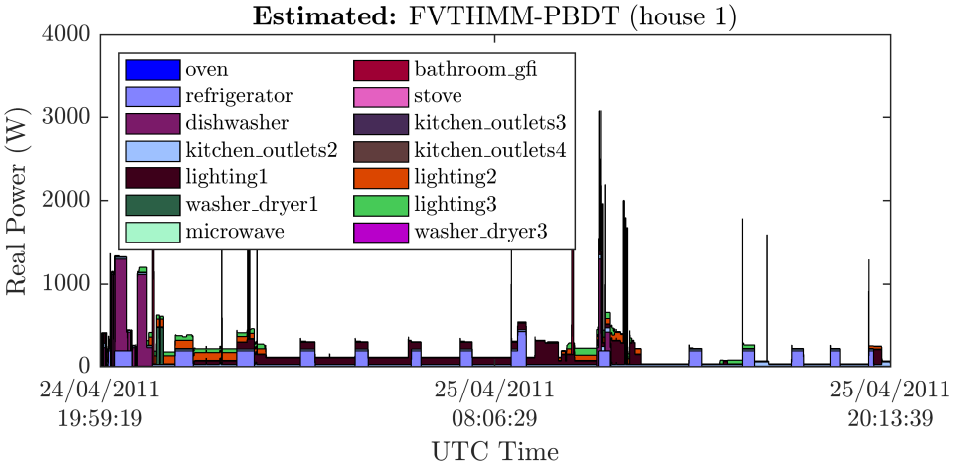



Figure 4.19: Disaggregation accuracy of different methods when applied to the REDD dataset.

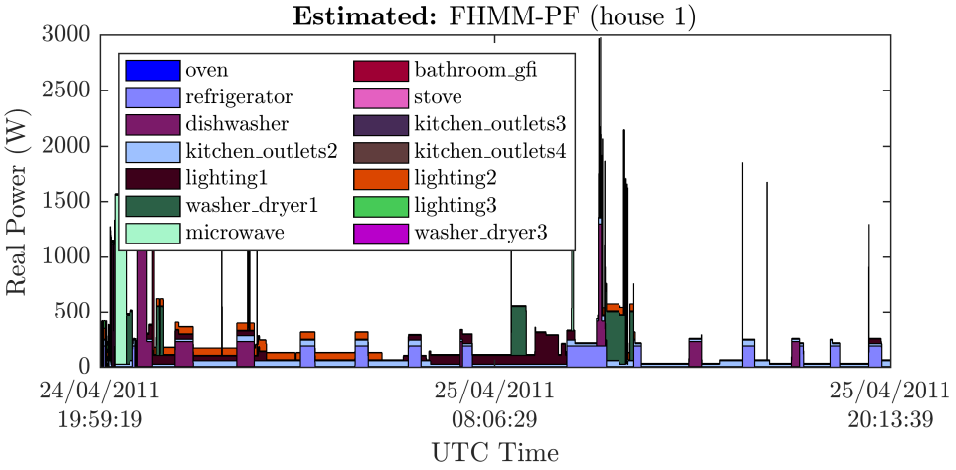
However, for house 2, FVTHMM-PF is found to be marginally better than FVTHMM-PBDT. A close inspection reveals a number of reasons. First, house 2 has the least number of submeters and the lowest $M_{\text{sys}}$ among all houses. Therefore, for any given $\mathbf{x}_t$, the corresponding $y_t$ has a lower variance on average. Consequently, the likelihood $p(y_t \mid \mathbf{x}_t)$ is more peaky, and sampling incrementally from the proposal distribution of PF, $p(\mathbf{x}_t \mid \mathbf{x}_{1:t-1}, y_t)$, is more likely to give $\mathbf{x}_t$ that is locally optimal for a given time slice but not optimal over a range of observation sequence. This scenario is evident in Figure 4.21 where the estimated state sequence arising from FVTHMM-PF turns out to have a lower log likelihood than that of FVTHMM-PBDT, even though the former's estimate is closer to the true state sequence on average. Secondly, as can be seen in Figure 4.22, the state estimates from FVTHMM-PBDT have noticeably higher false negatives for **`lighting`** and higher false positives for **`refrigerator`**, in comparison to FVTHMM-PF. The discussion that follows highlights all the other interesting re-




(a) The ground truth for a day's worth of data from house 1 of the REDD dataset.




(b) Estimated using FVTHMM-PBDT




(c) Estimated using FHMM-PF [EBE15]

Figure 4.20: Comparison between FVTHMM-PBDT and FHMM-PF in disaggregating one day's worth of data from house 1 of the REDD dataset.

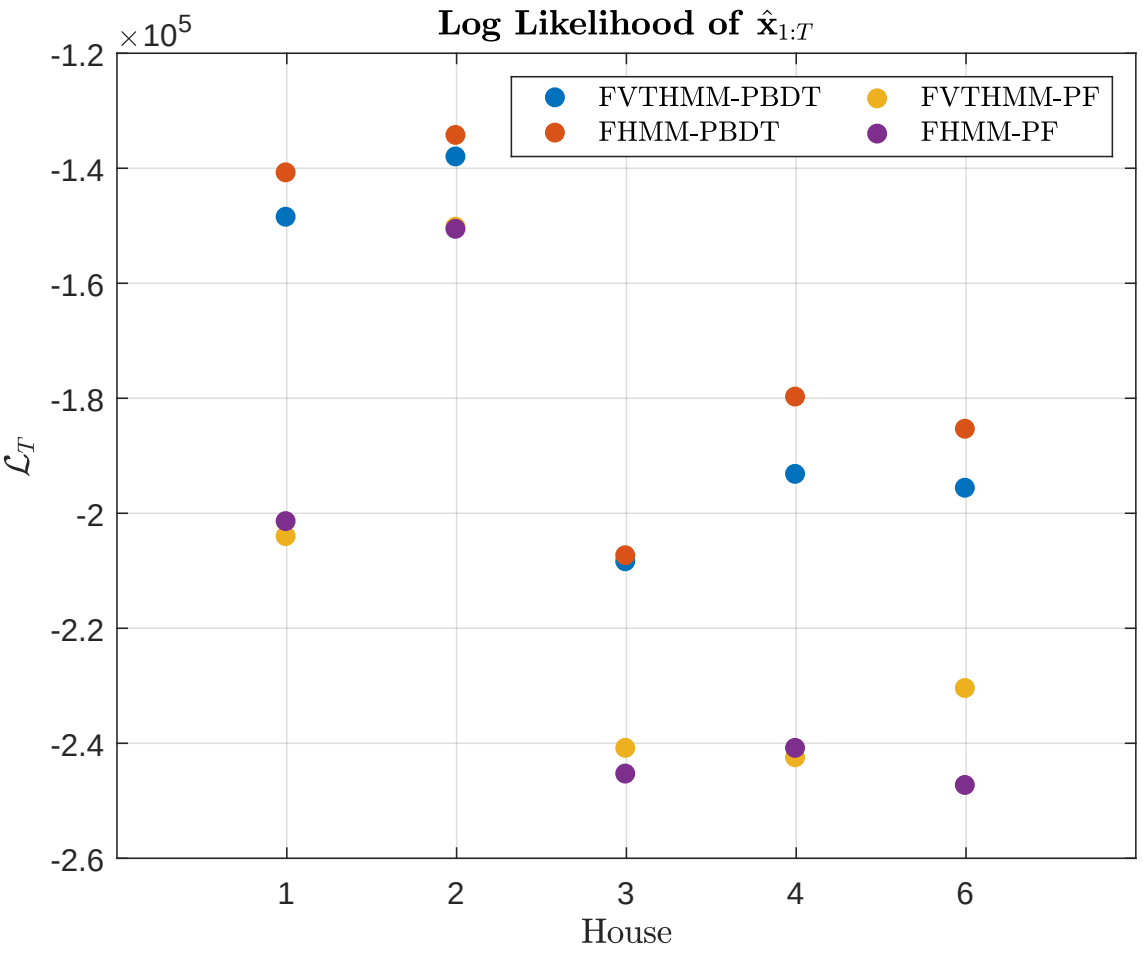



Figure 4.21: Log likelihood of the estimated state sequence for all test sets considered.

sults from the experiments. Subsequent to this, a summary of the cause of disaggregation errors and ways to rectify them are provided.

### Similarities between FHMM-PBDT and FHMM-Viterbi

In house 2, both FHMM-PBDT and FHMM-Viterbi have exactly the same disaggregation accuracy and the same estimated states, further confirming the previous claim that the PBDT algorithm is able to find the optimal $\hat{\mathbf{x}}_{1:T}$ when given a large enough $N_{\text{p,max}}$. While this also shows that a $N_{\text{p,max}}$ of 100 is clearly sufficient for the case of house 2 with FHMM, a higher $N_{\text{p,max}}$ is expected to be required for all other cases, especially methods utilising the FVTHMM model and houses with a large $M_{\text{sys}}$.

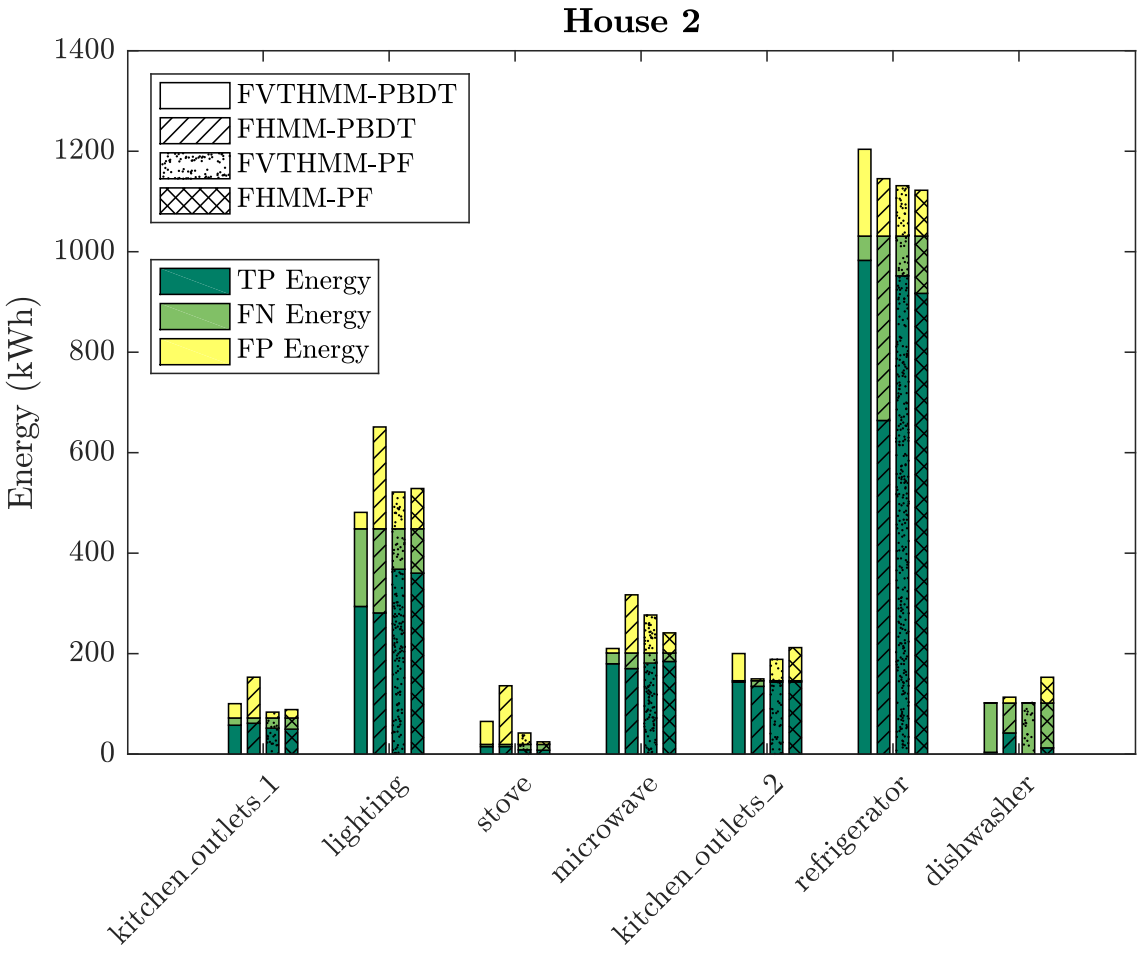
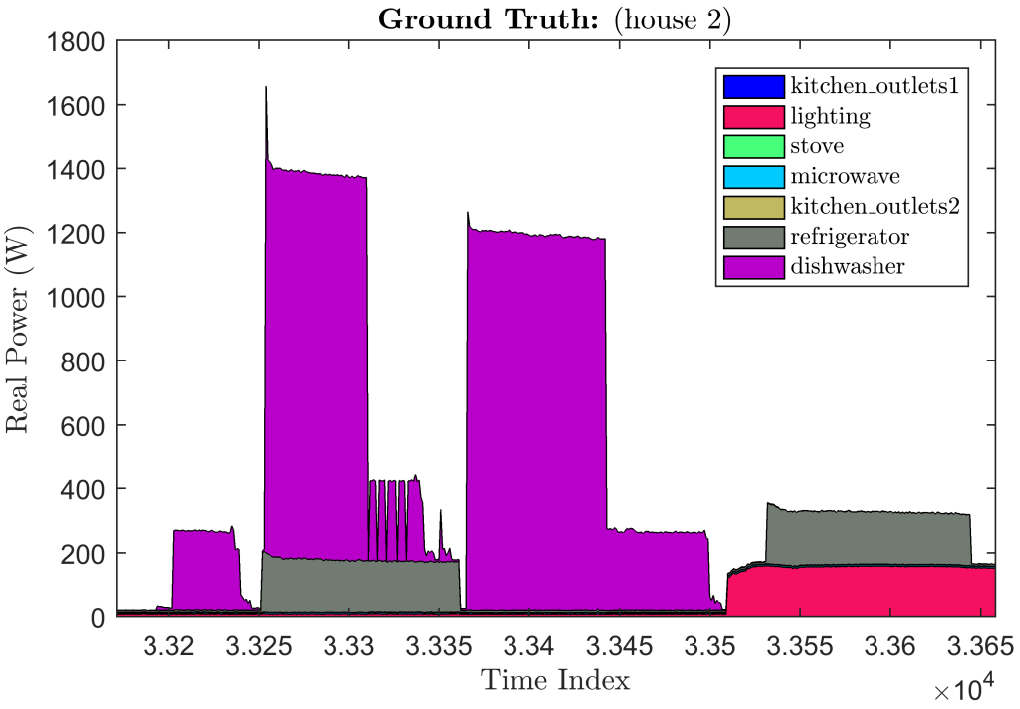



Figure 4.22: The energy associated with the true positives, the false negatives and the false positives for the test set of house 2.

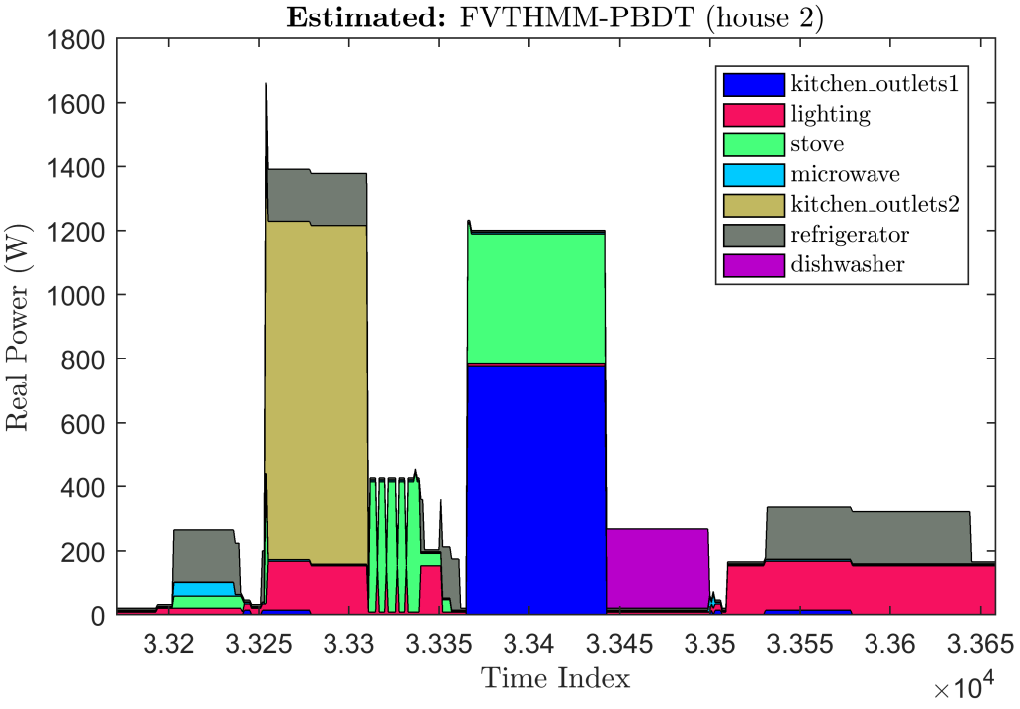
### Dishwasher problem of FVTHMM in house 2

As can be seen in Figure 4.22, regardless of the state estimation algorithm used under FVTHMM, the dishwasher in house 2 has high false negatives. The dishwasher is being misclassified as other appliances most of the time. An example of this is shown in Figure 4.23 where the dishwasher is supposed to be detected as being turned ON at $t = 33203$. Further analysis reveals a number of interrelated factors giving rise to this.

First, it was found that none of the $N_{\text{p,max}}$ particles at one time step before the dishwasher is turned ON (i.e. $t = 33202$) has the correct counter values; they all have $c_{t-1,\text{diswasher}}$ of 7376 instead of the true value of 20654. Therefore, regardless of the parent particles, this causes the corresponding duration-dependent state transition probability and by extension, the score of all generated particles with the correct dishwasher state to be low, given that it is not likely for a dishwasher to be turned on after only being turned off for 7376 time steps (see Figure 4.24). The




(a) The ground truth for a segment of data from house 2 of the REDD dataset.




(b) The same segment but with estimates from FVTHMM-PBDT.

Figure 4.23: Misclassification of **`dishwasher`**.

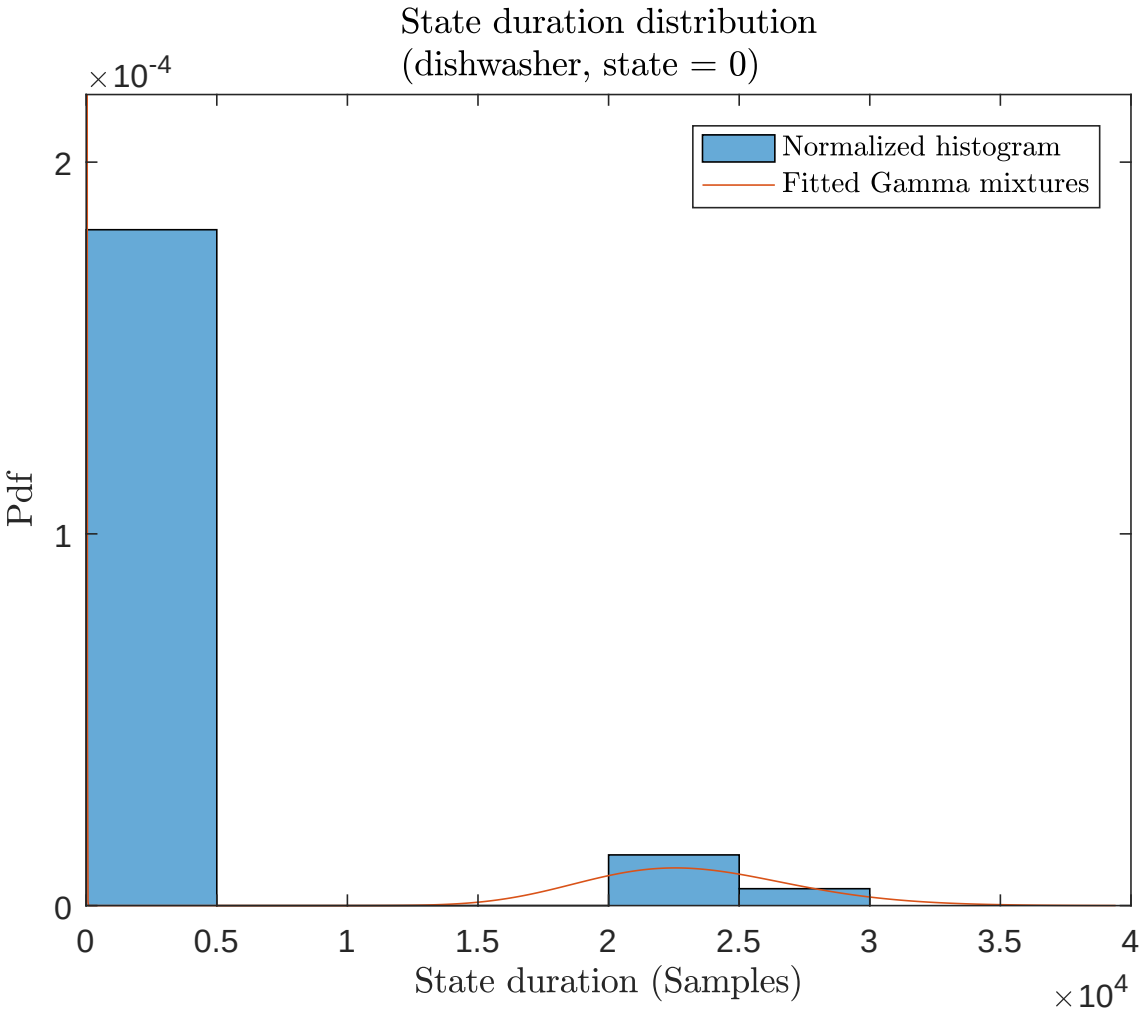
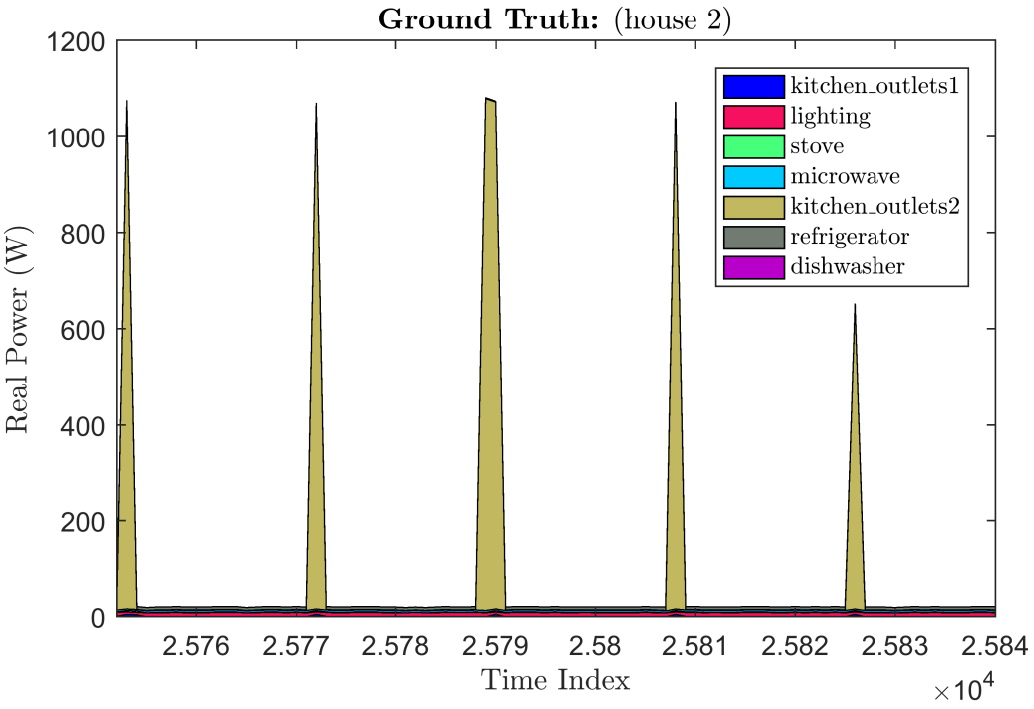


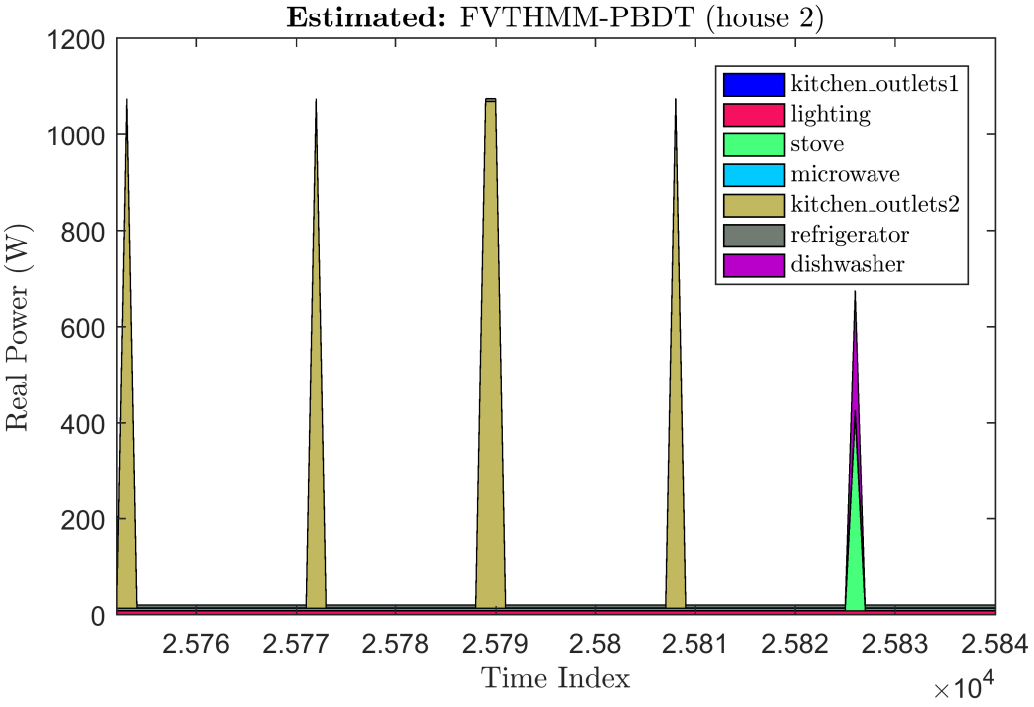

Figure 4.24: State duration distribution associated with the OFF-state of the dishwasher of house 2.

fact that all of the parent particles have counter values of 7376 suggests that the dishwasher is mistakenly detected to be ON at $t = 25826$. Figure 4.25 illustrates that this is indeed the case.

It turns out that the wrong state inference at $t = 25826$ is caused by a spurious observation (i.e. the power consumption deviates from what the model expects), leading to the case where both the stove and dishwasher are being erroneously assigned to the wrong state. If we consider the ground truth signal for the segment shown in Figure 4.25a, it can be seen that one of the pulses belonging to **`kitchen_outlets2`** is lower than usual, taking the value of about 670W. As the **`kitchen_outlets2`** is only modelled using two states with mean powers of about 0W and 1000W, the likelihood of observing such a value is extremely low. In fact, it is entirely reasonable to think that such an observation is not actually part of the operation of the **`kitchen_outlets2`**. The spurious observation could




(a) The ground truth for a segment of data from house 2 of the REDD dataset.




(b) The same segment but with estimates from FVTHMM-PBDT.

Figure 4.25: Misclassification of **`kitchen_outlets2`**

merely be an artefact of the sampling process where the value between transition from one state to another happened to be sampled before the transition is completed. The rare occurrence of observing this value in the **kitchen_outlets2** data supports this hypothesis.

Given that the pulses are almost periodic, one might expect the state duration part of the model to come in play so as to regulate the score of the true particle at $t = 25826$ from dropping too much. However, because the spurious observation falls in the tail of the Gaussian distributions corresponding to the two states of **kitchen_outlets2** and thus, able to influence more of the overall score of the particle with the true **kitchen_outlets2** state, a more likely but wrong state is given a higher score under the model.

Judging from all of this, it may seem that the misclassification of **kitchen_outlets2** at $t = 25826$ is largely responsible for the misclassification of **dishwasher** at $t = 33203$. To test whether this is true, we forced the counter value of the dishwasher just before it is actually turned on to its true value. Then, the disaggregation is resumed as it is. It turns out that after doing so, the dishwasher is now being correctly inferred to be ON at $t = 33203$. This is shown in Figure 4.26, validating our claim that the misclassification of the **dishwasher** is indeed due to the spurious observation at an earlier time step. In addition, when

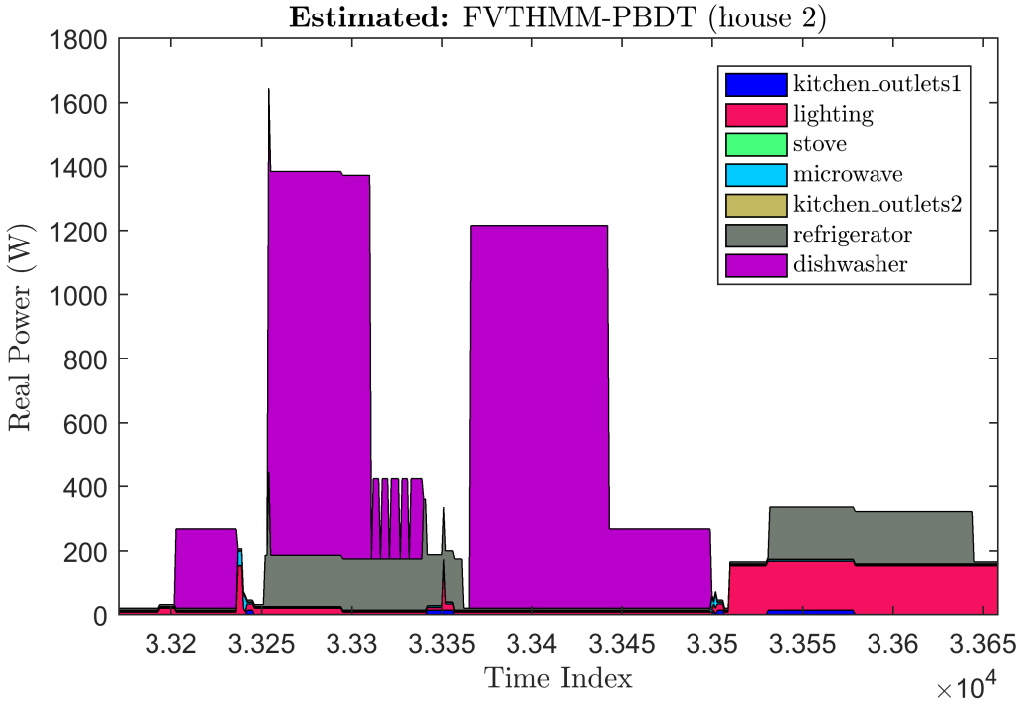



Figure 4.26: The estimated power after forcing the counter of the dishwasher to be the correct value, disregarding the effect of the spurious observation.

the dishwasher counter value is enforced, the CAR metric of house 2 increases to 86.91%, surpassing that of the FVTHMM-PF and the original CAR.

Also interesting is the observation that the CELLR of the error segment with the misclassified pulse has a positive value of +600.0, while the $\text{CELLR}_\text{e}$ and $\text{CELLR}_\text{d}$ for the same error segment are +624.3 and -24.23 respectively. All three CELLR metrics give credence to the claim that the error is model-induced, with both the $\text{CELLR}_\text{e}$ metric and the $\text{CELLR}_\text{d}$ metric confirming the emission model's inability to account for the spurious observation associated with **`kitchen_outlets2`**.

#### Refrigerator transient problem

A common trend across all houses considered is the misclassification that happens at the onset of the refrigerator's ON cycle. As an illustrative example, Figure 4.27b depicts the situation in house 1 where the power surge at the onset causes not just the **`refrigerator`** to be detected but also **`lighting1`**. This is largely attributed to the transient behaviour of the refrigerator not being taken into account by the refrigerator's emission model.

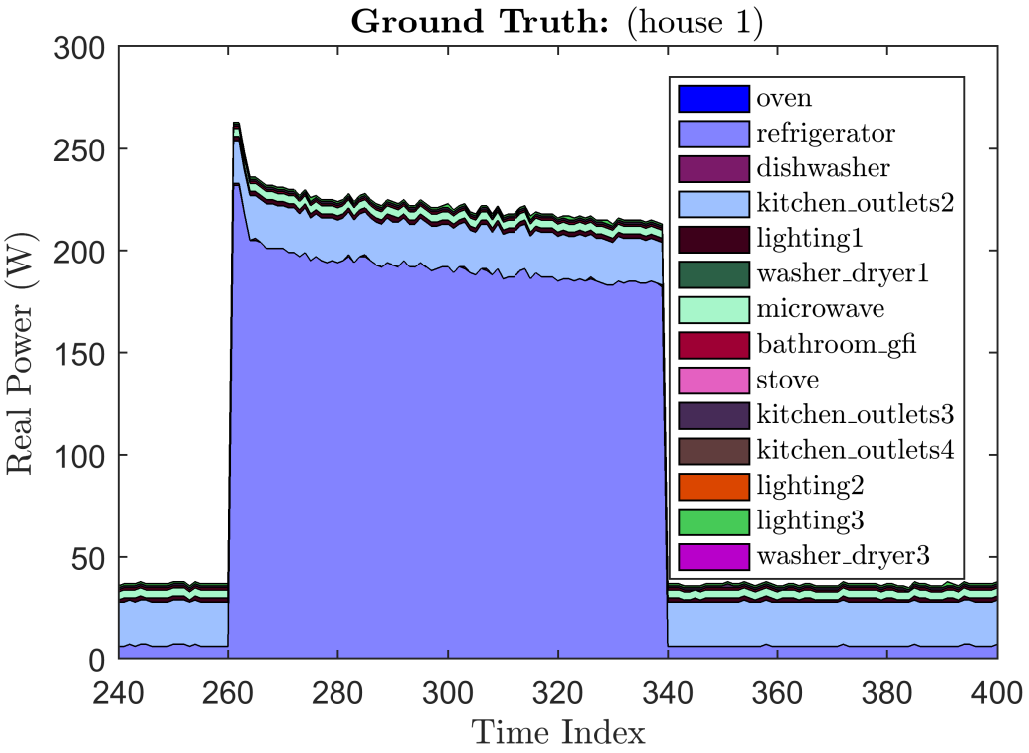
The empirical cumulative distribution functions (ECDF) of all CELLR metrics at segments where such errors occured are presented in Figure 4.28, Figure 4.29 and Figure 4.30. Only about 10% of the CELLR values are negative, strongly indicating that the errors are predominantly due to the shortcomings of the model used for state estimation instead of the particle truncation operation that is part of the PBDT algorithm.

Note also that 100% of the $\text{CELLR}_\text{e}$ values are positive while about 90% of the $\text{CELLR}_\text{d}$ values are negative. Together, these point to the insufficiency of the emission model being largely responsible for the errors.

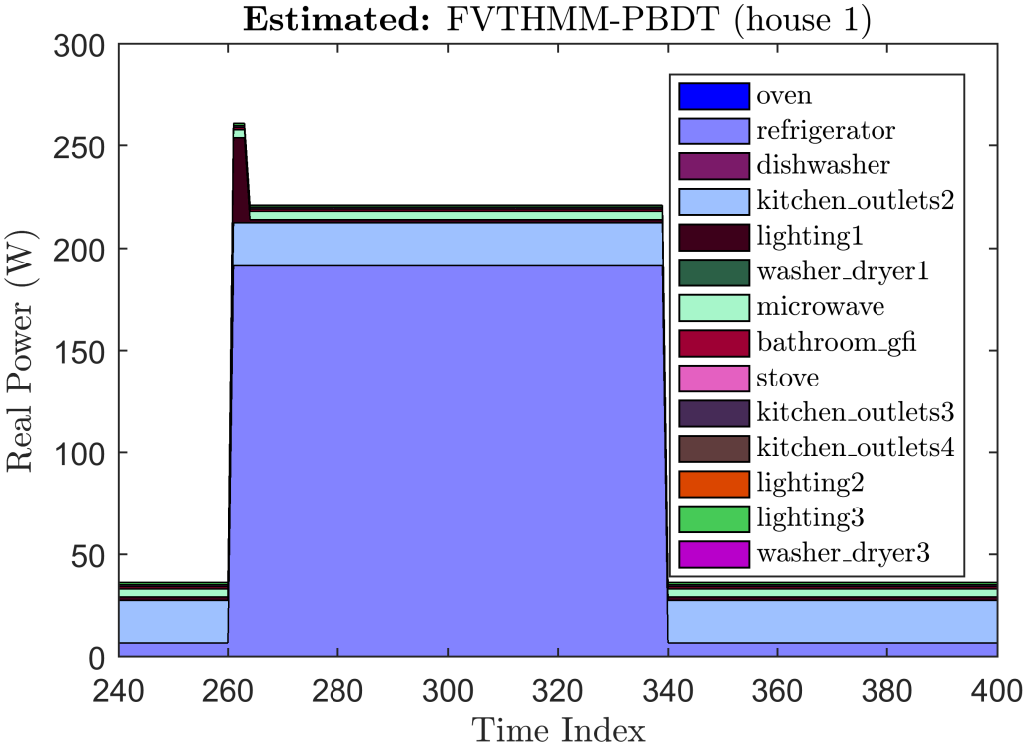
#### Poor disaggregation accuracy for all algorithms in house 4

Although Table 4.2 shows that the proposed method, FVTHMM-PBDT, outperforms the other methods in house 4, all algorithms have CAR values below 65%. The main reason for this is that appliances of house 4 appear to have large fluctuations on average and some of them do not conform well to the piece-wise constant model. This suggests that it is more challenging for the emission model to represent the varying power consumption for a given state more accurately.

An initial investigation reveals that the average number of states across all 11 submeters in house 4 is 5.36, which is the largest among all the other houses. It




(a) The ground truth for a segment of data from house 1 of the REDD dataset.




(b) The same segment but with estimates from FVTHMM-PBDT.

Figure 4.27: A depiction of the refrigerator transient problem.

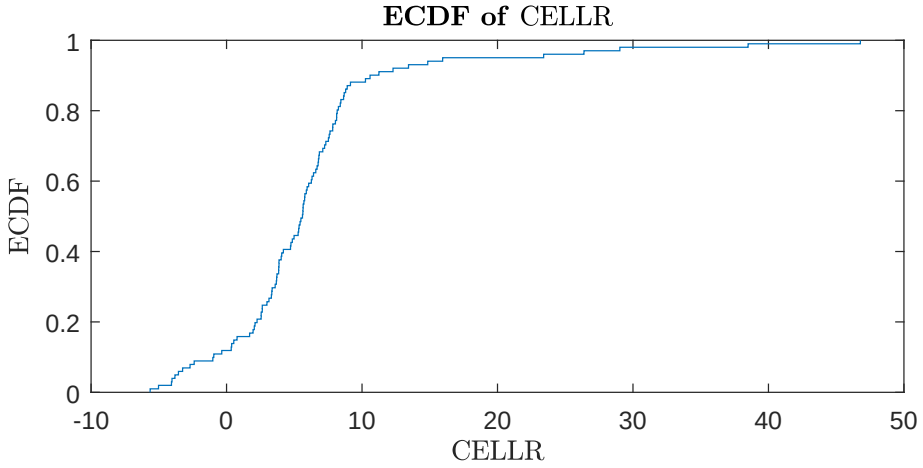



Figure 4.28: ECDF of CELLR for errors relating to the refrigerator transient problem in house 1 of the REDD dataset.

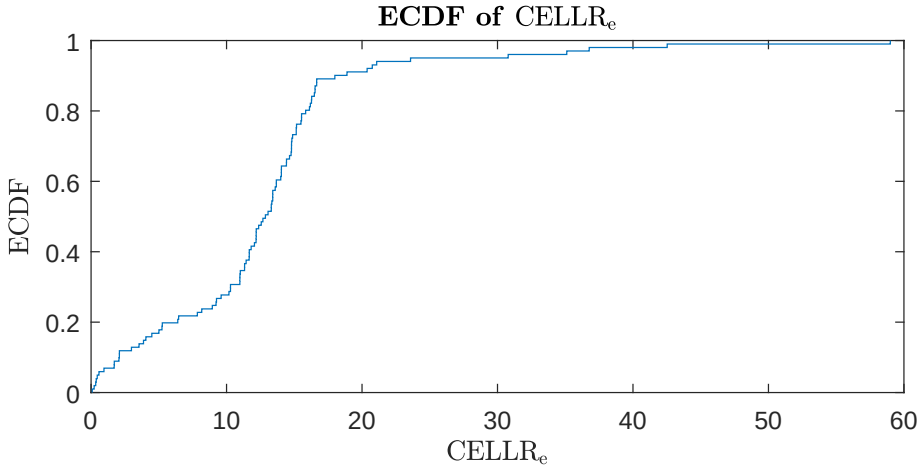



Figure 4.29: ECDF of CELLR$_e$ for errors relating to the refrigerator transient problem in house 1 of the REDD dataset.

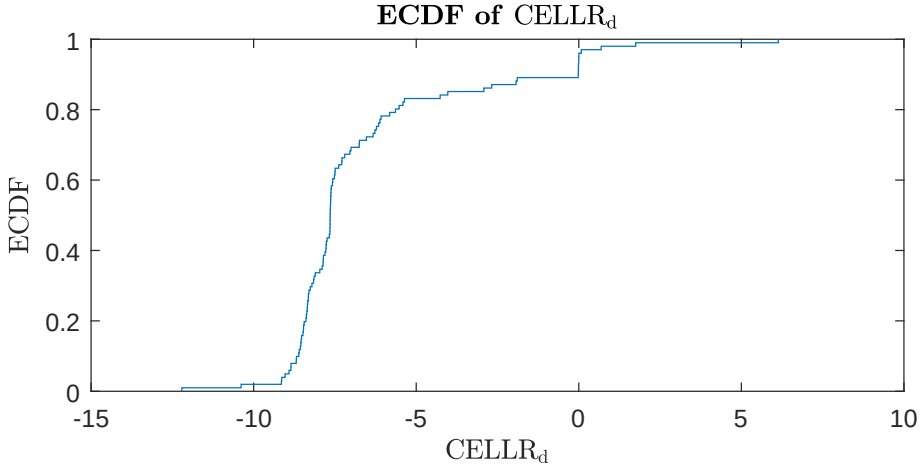



Figure 4.30: ECDF of CELLR$_d$ for errors relating to the refrigerator transient problem in house 1 of the REDD dataset.

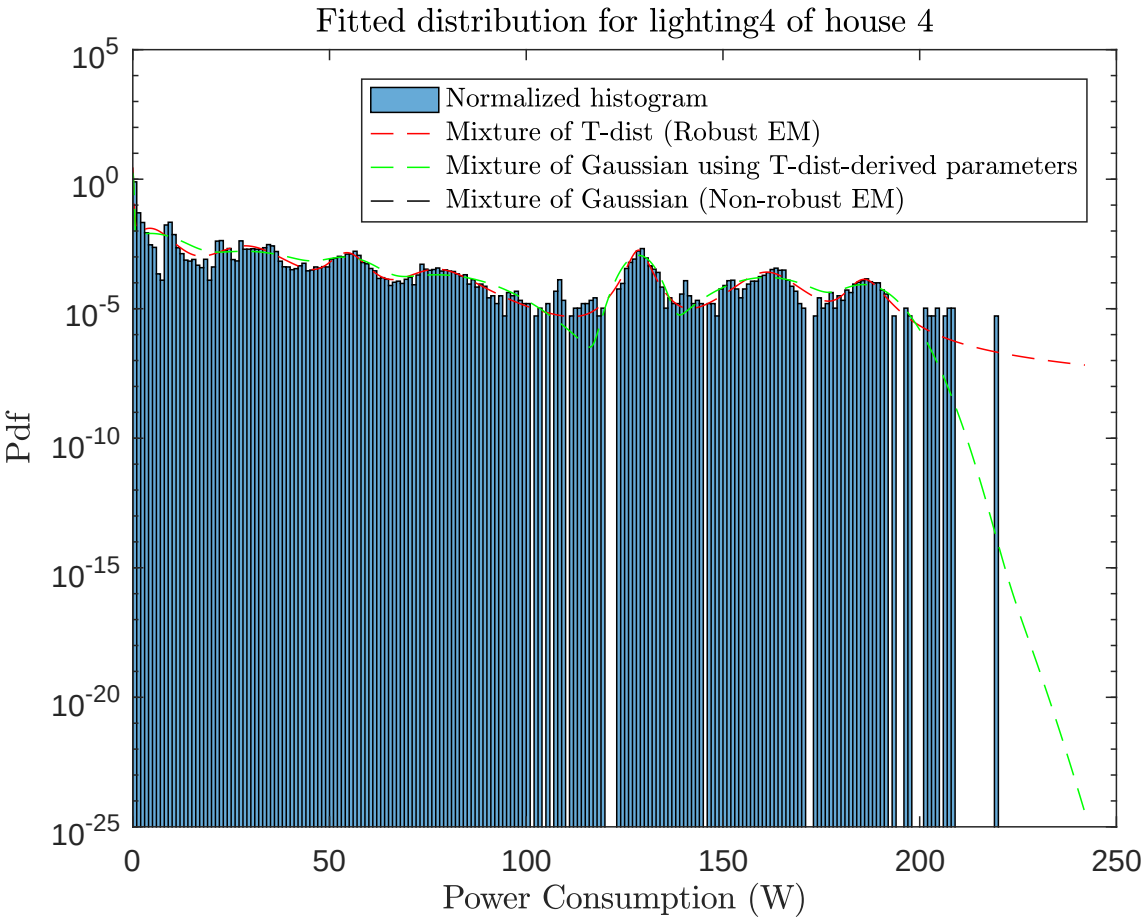



Figure 4.31: Emission model for **`lighting4`** of house 4 in the REDD dataset

was also found that there are 8 submeters (out of 11) having 5 or more states, meaning 73% of all submeters in house 4. In contrast, both house 1 and house 2 only have 14% of their submeters, while house 3 and house 6 have 41% and 62% respectively.

The high number of states used to model the majority of the appliances in house 4 can be attributed to the inherently large number of peaks in the power histogram of each appliance. Also, the lack of clearly defined peaks in certain histograms makes fitting the right emission model difficult. This can be seen in Figure 4.31 where the histogram of **`lighting4`** is illustrated.

In short, while FVTHMM-PBDT still leads the other methods, the high number of states and the inaccuracies in modelling the power consumption play a major part in the overall low disaggregation performance of house 4.

**Summary**

From the results presented thus far, the overarching theme appears to be that majority of the errors can be traced back to inaccuracies of the appliance emission models. There are a few ways to rectify this.

The first approach is to incorporate a more complex emission model which is able to more accurately account for the transient behaviours and the observed aggregate measurements that are not exactly piece-wise constant in nature. A specific instance of this will be briefly explored in Section 4.5.

Optionally, we can also include a preprocessing phase so that only steady-state power segments are extracted for disaggregation. The processed signal would be cleaner in the sense that the fluctuations are even out, allowing the PBDT algorithm to perform better. This will be discussed as part of the robustification extension in the next chapter.

Yet another way is to add an outlier detection to disregard spurious observations, preventing disaggregation of such values from propagating errors forward and subsequently affecting inferences of states negatively.

### 4.4.7 Empirical Analysis on Time Complexity

Apart from the disaggregation accuracy discussed in the previous subsection, the time complexity of the proposed method is also explored. We begin by investigating the relationship between the maximum number of particles to keep at each time step, $N_{\text{p,max}}$, and the runtime of FVTHMM-PBDT, before studying the variation of its runtime with the number of possible system states, $M_{\text{sys}}$, and the number of appliances, $K$. The discussion that follows presents the outcome in these two aspects.

**Runtime *vs* $N_{\text{p,max}}$**

To investigate the value $N_{\text{p,max}}$ and how it affects the runtime of FVTHMM-PBDT, the aggregate data from house 2 is disaggregated with $N_{\text{p,max}}$ of 1 and then from 10 to 100 in increments of 10. The associated runtime for each $N_{\text{p,max}}$ is recorded.

Shown in Figure 4.32 is the effect of $N_{\text{p,max}}$ on the runtime of the algorithm. The figure illustrates that for a fixed $M_{\text{sys}}$ and a fixed $T$ of 50000, the runtime increases linearly with $N_{\text{p,max}}$. This is consistent with the theoretical worst-case time complexity of $O(M_{\text{sys}}N_{\text{p,max}}T)$.

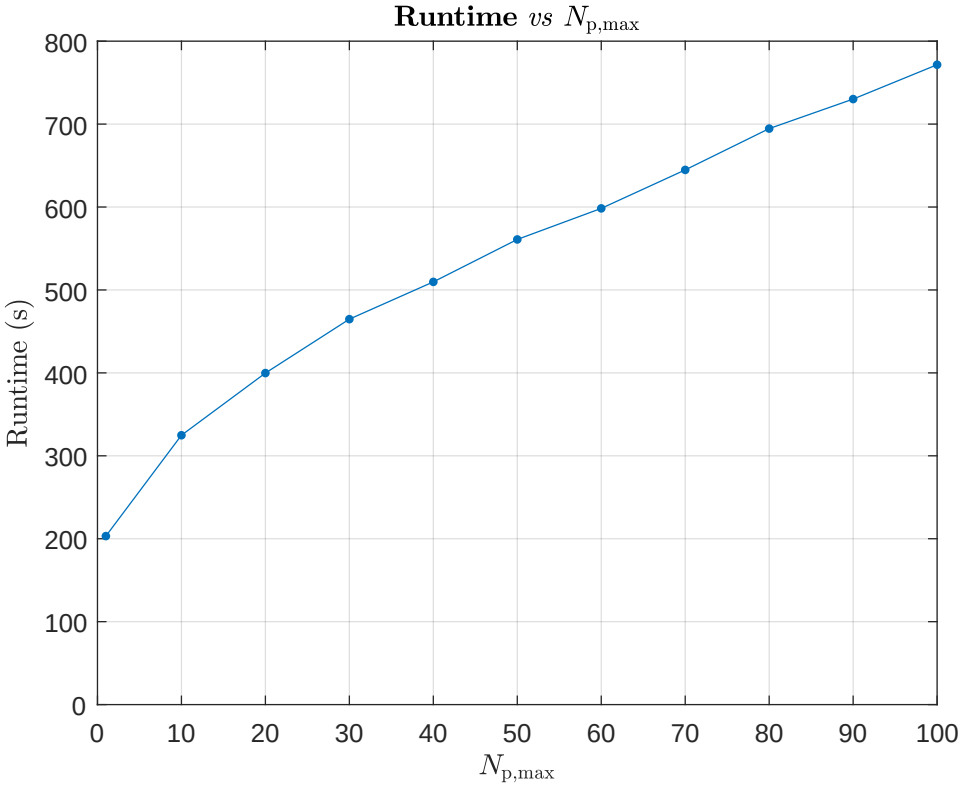



Figure 4.32: Runtime of FVTHMM-PBDT *vs* $N_{\text{p,max}}$.

**Runtime *vs* $M_{\text{sys}}$ and $K$**

In this test, we explore the role of $M_{\text{sys}}$ and $K$ in influencing the runtime of the algorithm. All houses considered for the analysis thus far are included for the investigation except for house 2, given that it only contains a very small number of appliances. To vary the value of $M_{\text{sys}}$ for each house, we select $K$ submeters out of $K_{\text{max}}$. As there are $\binom{K_{\text{max}}}{K}$ combinations of $K$ submeters, only the one that results in the largest $M_{\text{sys}}$ is chosen. The power consumption of the chosen $K$ submeters are then added together to form the test aggregate measurements that will be disaggregated via FVTHMM-PBDT with $N_{\text{p,max}}$ of 100. We perform the experiment using $K$ from 1 to $K_{\text{max}}$, and the runtime for each $K$ and the resulting $M_{\text{sys}}$ is noted.

The results are summarised in Figure 4.33 and Figure 4.34, where it is shown that the algorithm runtime appears to increase approximately logarithmically with $M_{\text{sys}}$ and approximately linearly with the number of appliances/submeters, $K$. Clearly, this is a marked improvement over the theoretical worst-case time complexity of a naive implementation of PBDT (i.e. $O(M^K N_{\text{p,max}} T)$ if each appliance has $M$ states) and the theoretical time complexity of the Viterbi algorithm under FVTHMM (i.e. $O(M^{2K} T^{2K+1})$), thus demonstrating the effectiveness of the

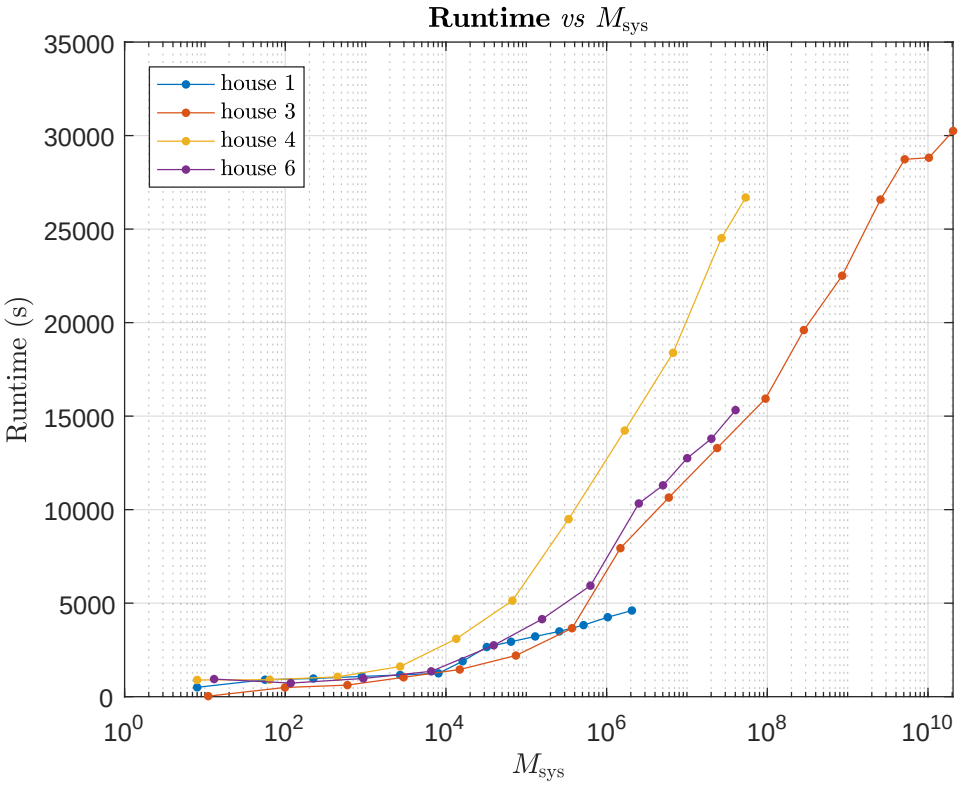



Figure 4.33: Runtime of FVTHMM-PBDT *vs* $M_{\text{sys}}$.

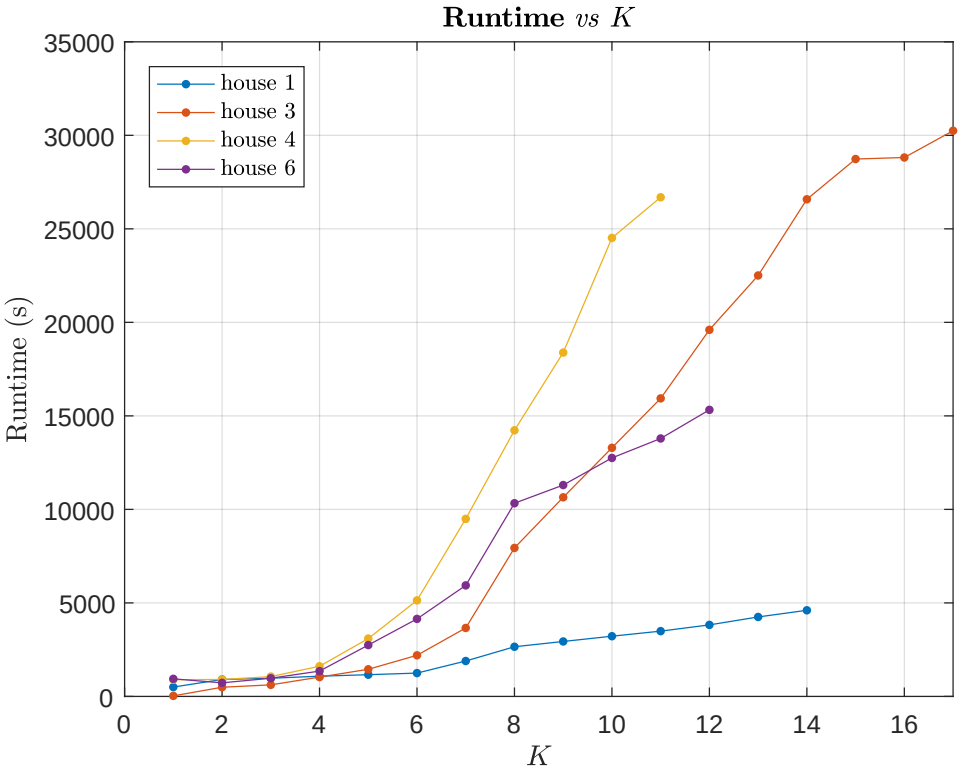



Figure 4.34: Runtime of FVTHMM-PBDT *vs* $K$.

state-pruning stage detailed in Section 4.3.1 and the computation-sharing scheme described in Section 4.3.2. The former partially decouples the number of system states to consider from the actual $M_{\text{sys}}$ while the latter reduces the number of computations as arising from each parent particle by sharing the calculation within a group. In most cases, the distribution of the particles at each time step is easily exploitable for computation-sharing and the speed-up as a result of this can be substantial.

The different gradients of the runtime curve for different houses as shown in both figures might be due to the nature of how the appliances contributes to the aggregate measurements in each house. As noted in the preceding subsection, house 4 in particular has appliances with less well-defined power levels that are more challenging to model. Consequently, there may be more uncertainty in each of the particles' estimate and computations are more difficult to be shared across particles, resulting in a runtime that increases more as $K$ grows. In contrast, the appliances in house 1 are simpler to model since they have more distinctive power levels. As such, it may be easier for the algorithm to group particles for computation-sharing given a particular observed aggregate measurement.

The figures also highlight that, in a system with on the order of 2 million states (i.e. house 1), the time needed to process 50000 samples is only about 4600 seconds (even in an interpreted language like MATLAB). On average, this means the time needed to process each sample is 0.092 seconds, resulting in an average throughput of 10.87 samples/s. As the samples are approximately 10 seconds apart, FVTHMM-PBDT is able to process faster than the rate at which new measurements arrive, thereby, supporting the claim that the proposed method is able to meet the demands of real-time computation for the purpose of NILM.

Adding to that is the observation that house 3 has 20 billion states and it takes the FVTHMM-PBDT method 35213 seconds to process 50000 samples. Although the increase in runtime over that of house 1 is expected, the average throughput of 1.42 samples/s is still beyond the minimum required throughput for achieving real-time computation. Moreover, it illustrates that increasing the cardinality of the system states by a factor of 10000 only penalises the average throughput by a relatively small factor of 7. Therefore, the PBDT algorithm can be considered a scalable method of inferring the hidden states of appliances, even when $M_{\text{sys}}$ is large.

For even greater speed improvements, the computationally independent relationship between the particles at each time step could be exploited to enable parallel executions of computations on a graphics processing unit (GPU) or a field-

programmable gate array (FPGA). However, such implementations are beyond the scope of this research and they can be conducted as part of any future work in the same direction.

### 4.4.8 Sensitivity Study on Sampling Intervals

The preceding experiments have all been done on data of sampling intervals of approximately 10 seconds. Considering that power measurements can arrived at different rates depending on the instrumentation hardware (e.g. smart meters), it is interesting to investigate the impact of different sampling intervals can have on the disaggregation accuracy of the proposed FVTHMM-PBDT.

To that end, we downsampled the original data from house 1 of the REDD dataset by different factors to simulate sampling intervals of 30 seconds, 60 seconds, 300 seconds and 900 seconds, then re-run FVTHMM-PBDT with $N_{\text{p,max}}$ of 100 for each of the different sampling intervals, while noting the resulting disaggregation accuracy during each run.

The outcome is shown in Figure 4.35. As can be seen, the CAR increases to a peak of 86% at 60 seconds before decreasing gradually as the sampling interval increases. The initial increase can be explained by the reduction in disaggregation errors as the transient effects are removed due to higher sampling intervals, while the subsequent decrease can be attributed to the increased likelihood of discarding samples that are important for the algorithm to make the right inferences.

Although this preliminary result appears to show that the proposed method is able to maintain high disaggregation accuracy up to sampling intervals of 900 seconds, more experimental results from other houses need to be obtained for a more definitive conclusion to be drawn. That said, it is a promising outcome.

## 4.5 PBDT with Segmental FVTHMM

We have seen in the previous section that one of the main contributors to state estimation errors is the inaccuracies of the appliance emission model. To that end, we will give a brief treatment of an augmented form of FVTHMM. Specifically, the appliance power is modelled to be dependent not just on its state but also on its counter value, i.e. $p(y_{t,k} \mid x_{t,k}, c_{t,k})$. The advantage is, it allows the decaying or the rising power consumption for a given appliance state to be represented naturally. While this representation has never been used for NILM before, it is

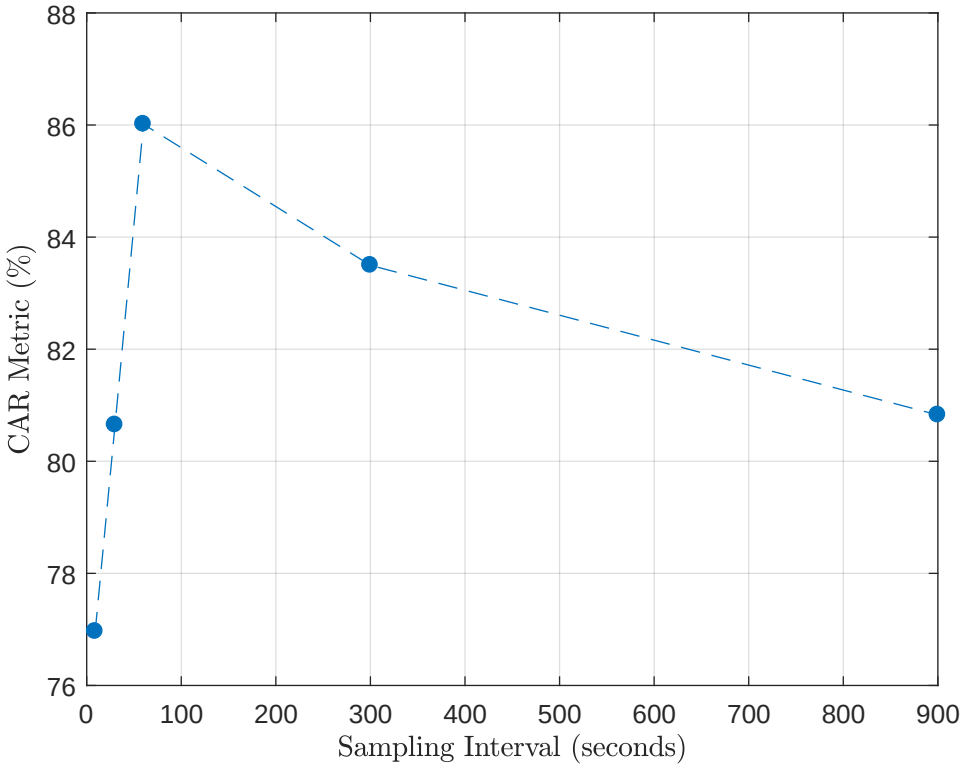



Figure 4.35: The impact of different data sampling intervals on the disaggregation accuracy of FVTHMM-PBDT. The result is obtained by applying the algorithm on house 1 of the REDD dataset.

closely related to the segmental hidden Markov model applied to other domains like speech modelling [KM91, Rus93, GY93].

Given the use of counter variables, the dependence of the power consumption on the dwell time is a natural and straightforward addition to FVTHMM, whereas such extensions are difficult to be made for the explicit-duration formulation of the hidden semi-Markov model as used by Kim et al. [KAL11] and Johnson and Willsky [JW13] for NILM, since the state duration distributions are directly employed in their work without the notion of time-varying duration-dependent quantities like in the variable-transition formulation, i.e. VTHMM.

In this section, the augmented model is formally described and only a short proof-of-concept is presented for demonstrating its successful application to resolve the transient issues associated with the operation of refrigerators. There is much scope for more work be done for exploring the best form of the segmental emission model $p(y_t \mid \mathbf{x}_t, \mathbf{c}_t)$. In the future, it is also hoped that the model be tested against a wider range of data to better evaluate its robustness.

### 4.5.1 Model Description

The model is similar to the original FVTHMM described in Section 3.3.1 of Chapter 3, except that the emission variable $y_t$ is also now conditionally dependent on a vector of counters. Formally, the joint probability over all the random variables is

$$\begin{aligned} p(\mathbf{x}_{1:T}, y_{1:T}, \mathbf{c}_{1:T}) = p(\mathbf{x}_1)p(\mathbf{c}_1) \prod_{t=1}^{T} p(y_t \mid \mathbf{x}_t, \mathbf{c}_t) \\ \times \prod_{r=2}^{T} p(\mathbf{x}_r \mid \mathbf{x}_{r-1}, \mathbf{c}_{r-1}) p(\mathbf{c}_t \mid \mathbf{x}_r, \mathbf{c}_{r-1}, \mathbf{x}_{r-1}), \end{aligned} \tag{4.24}$$

while its recursive form is

$$\begin{aligned} p(\mathbf{x}_{1:t}, y_{1:t}, \mathbf{c}_{1:t}) = p(\mathbf{x}_{1:t-1}, y_{1:t-1}, \mathbf{c}_{1:t-1}) p(y_t \mid \mathbf{x}_t, \mathbf{c}_t) \\ \times p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{c}_{t-1}) p(\mathbf{c}_t \mid \mathbf{x}_t, \mathbf{c}_{t-1}, \mathbf{x}_{t-1}). \end{aligned} \tag{4.25}$$

The dynamic Bayesian network (DBN) representation, illustrated in Figure 4.36, summarises the conditional independence assumptions of the model, with the directed connections between $y_t$ and $c_{t,k}$ shown to make the dependence explicit.

With consideration of the gradual decrease in appliance power shown in Figure 4.37, we have chosen to use the exponential relation linking the power consumption of appliance $k$ at time $t$, $y_{t,k}$, to its counter $c_{t,k}$ and its state $x_{t,k}$. That is,

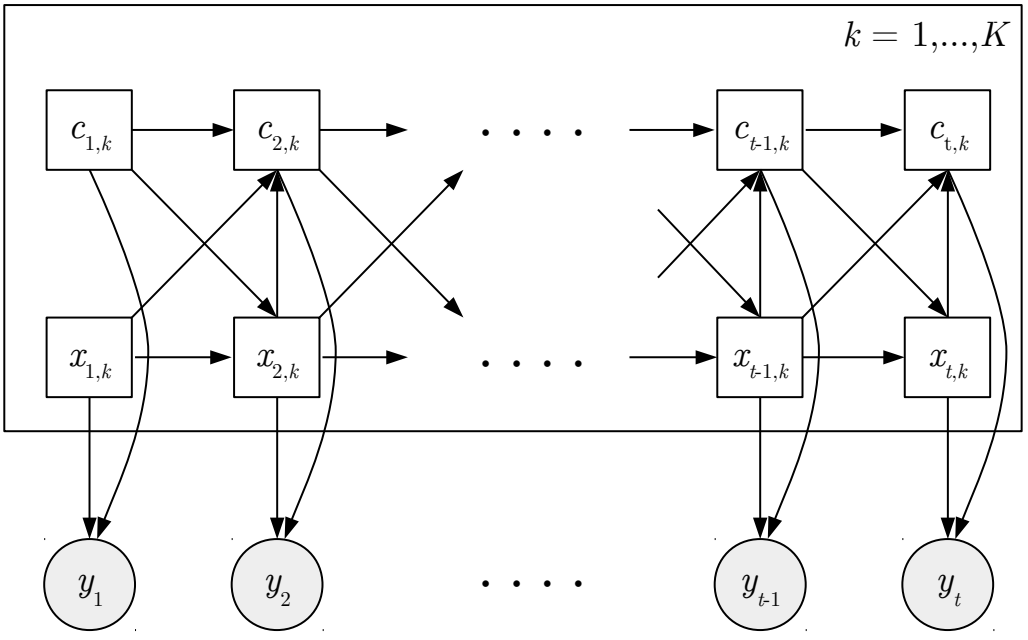



Figure 4.36: Dynamic Bayesian network representation of the Segmental FVTHMM.

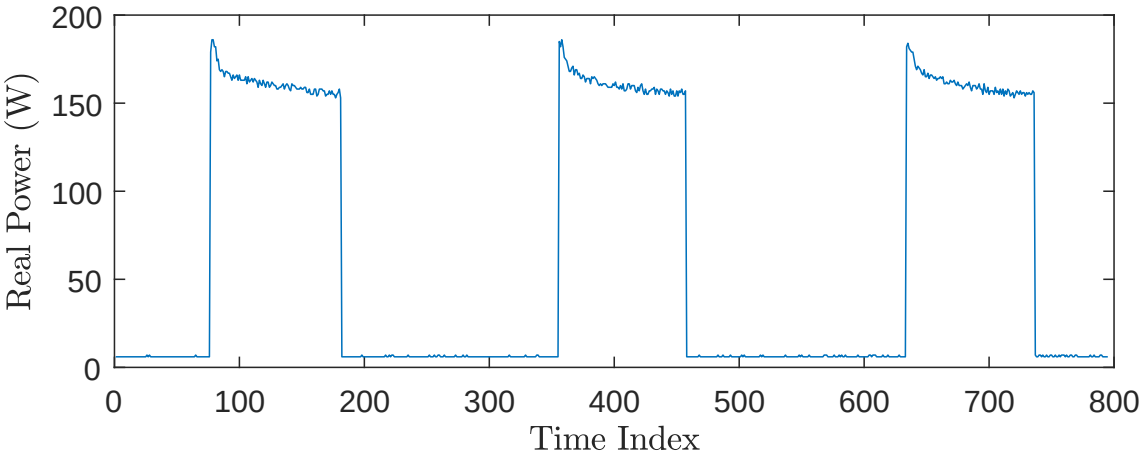



Figure 4.37: An example of the gradual decay in power consumption.

for a given state $x_{t,k} = i$ and a given counter value $c_{t,k} = \tau$,

$$y_{t,k} = \underbrace{a_i \exp(-b_i \tau) + \mu_i}_{f_k(i,\tau)} + n_k(i), \tag{4.26}$$

where $n_k(i)$ is the state-dependent noise term assumed to be distributed according to a zero-mean Gaussian distribution like before; $f_k(i, \tau)$ is the deterministic part of the power consumption that is a function of both the appliance state and the dwell time; $a_i$, $b_i$ and $\mu_i$ denote the parameters that should be fitted. Other forms of $f_k(i, \tau)$ could be used, and the noise term could also be a function of the dwell time. However, they are not considered for our demonstration; they are reserved for future work.

With the relation specified in (4.26), the power consumption of appliance $k$ at time $t$ is a random process characterised by

$$y_{t,k} \mid x_{t,k}, c_{t,k} \sim \mathcal{N}(a_{x_{t,k}} \exp(-b_{x_{t,k}} c_{t,k}) + \mu_{x_{t,k}}, \sigma^2_{x_{t,k}}). \tag{4.27}$$

In this way, for a fixed $x_{t,k}$, the mean power consumption is said to vary exponentially with $c_{t,k}$ while $\mu_{x_{t,k}}$ acts as a bias term which corresponds to the mean of the non-segmental version if $a_{x_{t,k}}$ is 0. As the noise term in (4.26) is independent of the counter, the variance is constant.

Now, if we assume that $y_{t,k}$ is not observable and only the aggregate power consumption, $y_t$, could be measured, the overall process becomes

$$y_t \mid \mathbf{x}_t, \mathbf{c}_t \sim \mathcal{N}\left( \sum_{k=1}^{K} a_{x_{t,k}} \exp(-b_{x_{t,k}} c_{t,k}) + \mu_{x_{t,k}}, \sum_{k=1}^{K} \sigma^2_{x_{t,k}} \right). \tag{4.28}$$

### 4.5.2 Parameter Estimation

In this subsection, the method used for inferring the parameters, $a_i$ and $b_i$, of the segmental emission model is outlined. The techniques used for estimating the parameters related to the state transition and the state duration are not presented in the following discussion, since they are exactly the same as those described in Chapter 3.

To determine the parameters for a given state $x_{t,k} = i$, the power data of appliance $k$ from the training set, $y_{1:T,k}$, is first segmented via the segmental $k$-means algorithm detailed in Section 3.4.2 of Chapter 3 to obtain the estimated state sequence, $\hat{x}_{1:T,k}$. Then, the power values corresponding to blocks or segments of consecutive $i$ in $\hat{x}_{1:T,k}$ are extracted; if there are $S$ segments, they are denoted by $\{y_{(u_s:v_s),k}\}_{s=1}^{S}$, where $u_s$ and $v_s$ signify the starting index and the ending index of segment $s$ respectively, with the underlying states satisfying $x_{u_s,k} = x_{u_s+1,k} = \ldots = x_{v_s-1,k} = x_{v_s,k} = i$ for all $s \in [1, S]$.

Next, all extracted segments are aligned such that their starting indices coincide, giving rise to a set of data points for each counter value, i.e. $\{y_{u_s,k}\}_{s=1}^{S}$ for $\tau = 1$, $\{y_{u_s+1,k}\}_{s=1}^{S}$ for $\tau = 2$ and so on. From this, $f_k(i, \tau)$ is estimated as the mean of $y_{u_s+\tau-1,k}$ for a given $\tau$, i.e. $\bar{f}_k(i, \tau) = \sum_{s=1}^{S} y_{u_s+\tau-1,k}/S$, before their computed values and their variation across $\tau$ are fitted using regression analysis. In the process, the parameters $a_i$, $b_i$ and $\mu_i$ are obtained.

As mentioned before, the variation of the variance across a segment is not incorporated. Therefore, we use the same variance as obtained using the techniques from Chapter 3. Exploring the impact of allowing the variance to vary is a promising direction for future work.

### 4.5.3 Segmental Modelling: An Example

As an example to the previous discussion, we will consider using the refrigerator from house 2 of the REDD dataset. Figure 4.38 illustrates its power consumption for one particular activation with two states, and it is clear that one characteristic common to both states is the slow decay in power consumption from the onset of the state transition. Indeed, the characteristic is frequent enough that when all such segments corresponding to each state are aligned, a distinct trend can be observed. The heat maps demonstrating this for state 1 and state 3 of the refrigerator are depicted in Figure 4.39 and Figure 4.40 respectively. Each horizontal line in the heat map shows the variation of power across one segment, while the

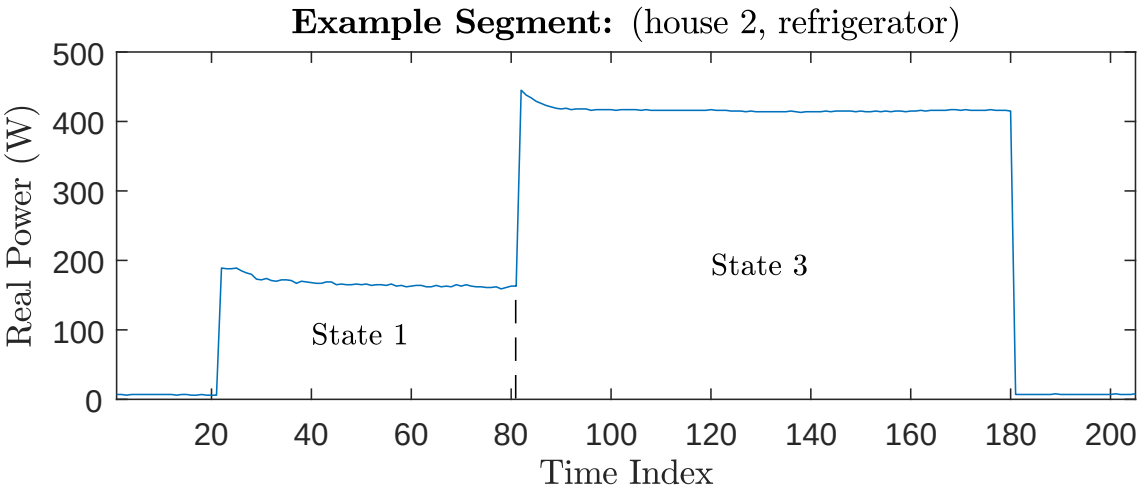



Figure 4.38: An example of an ON period with two states in the refrigerator of house 2 of the REDD dataset.

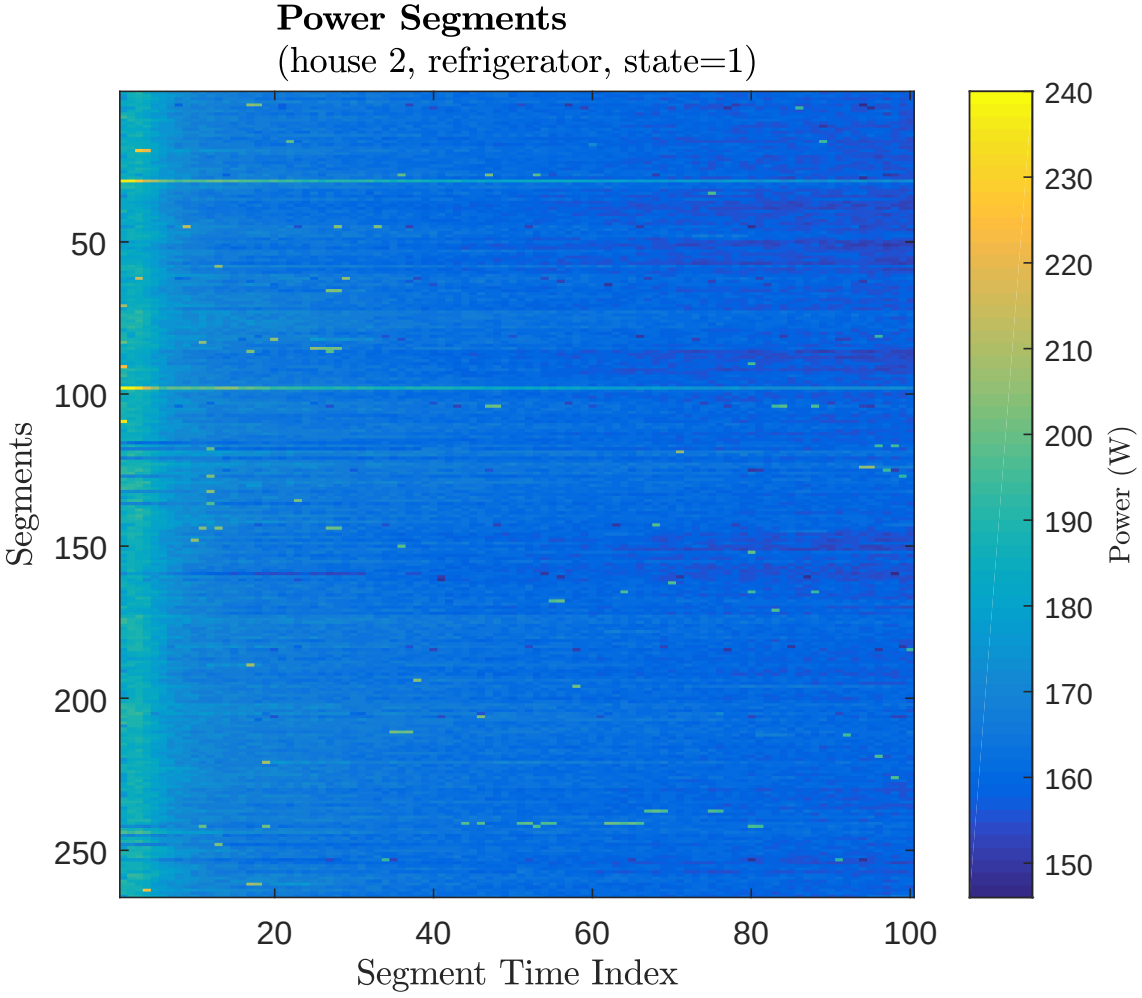



Figure 4.39: The variation in power for the first 100 time steps of the segments corresponding to state 1 of the refrigerator from house 2 of the REDD dataset.

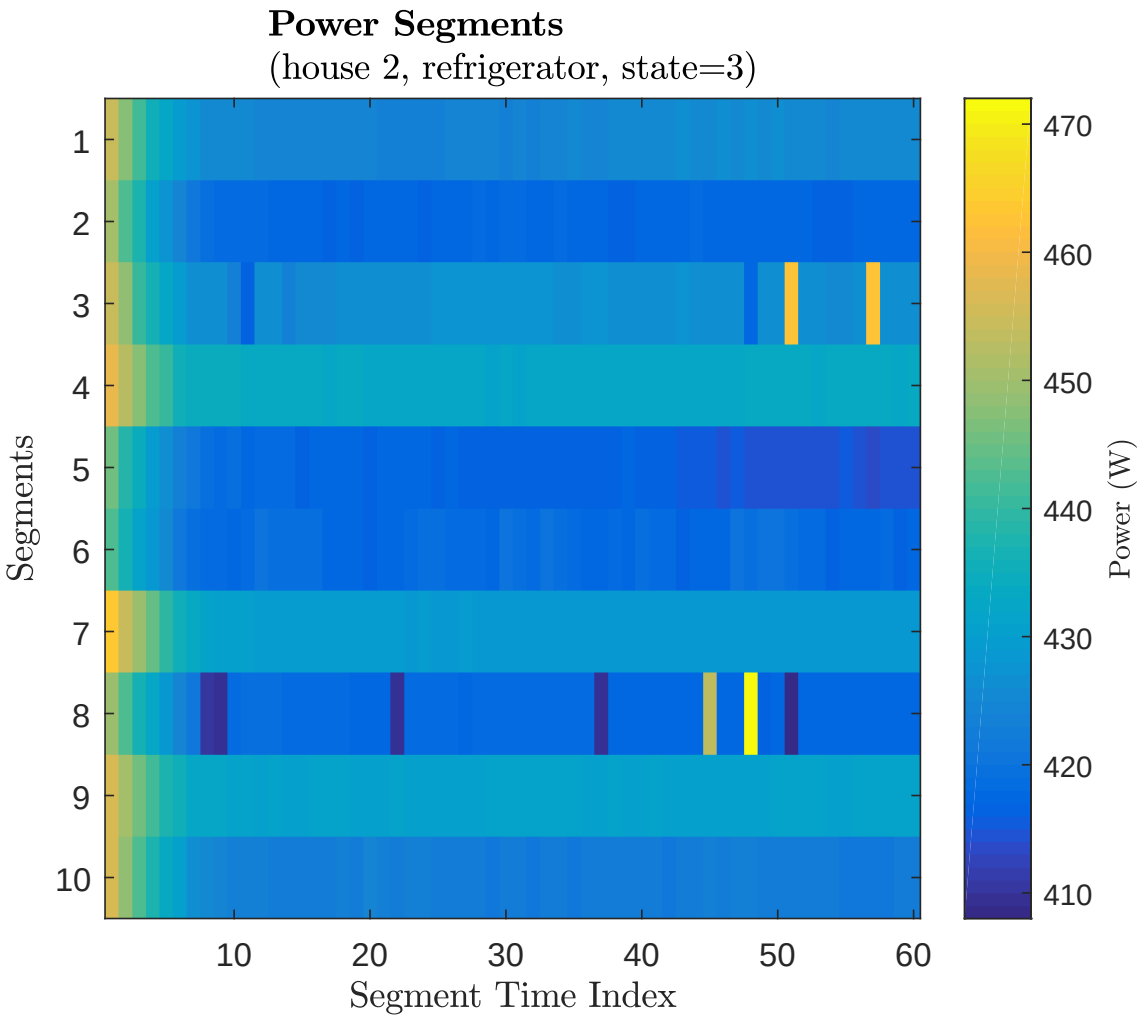



Figure 4.40: The variation in power for the first 60 time steps of the segments corresponding to state 3 of the refrigerator from house 2 of the REDD dataset.

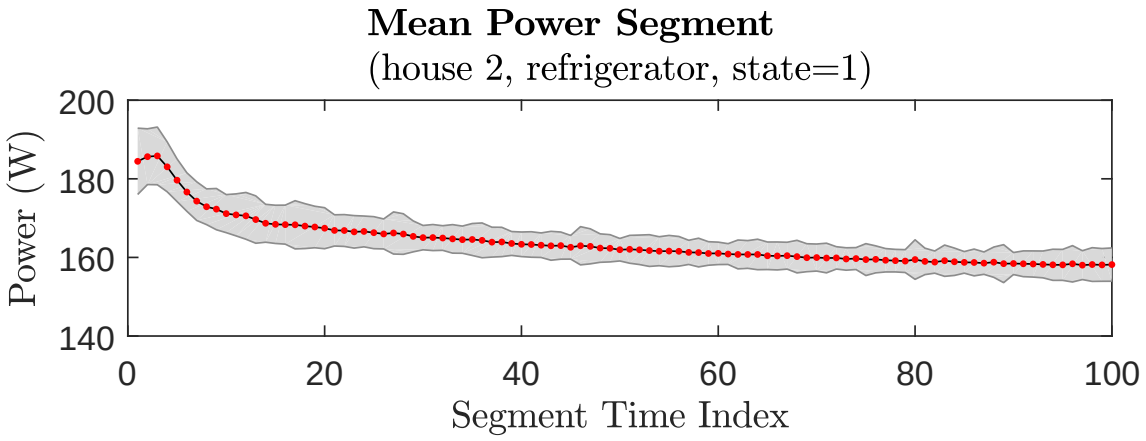



Figure 4.41: Mean power consumption values for the first 100 time steps of the segment corresponding to state 1. The distance of the error bar from the centre signifies the standard deviation.

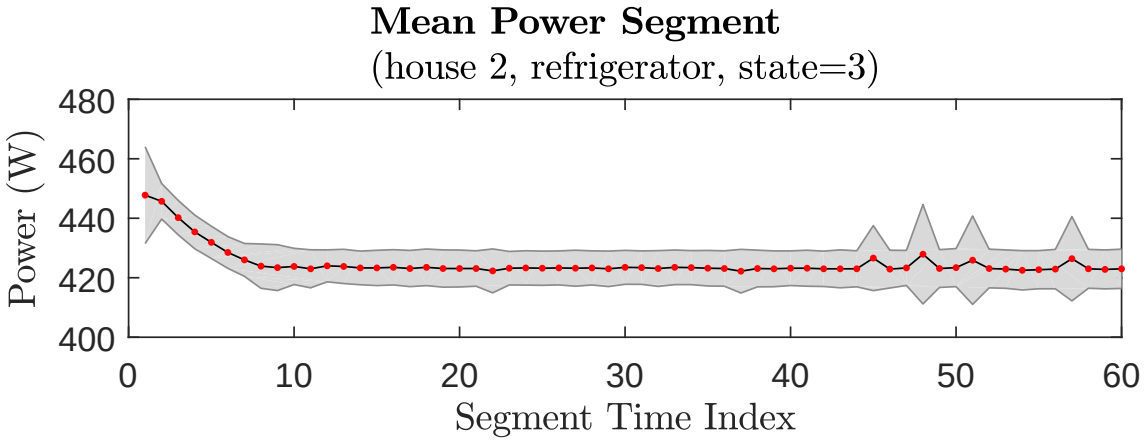



Figure 4.42: Mean power consumption values for the first 60 time steps of the segment corresponding to state 3. The distance of the error bar from the centre signifies the standard deviation.

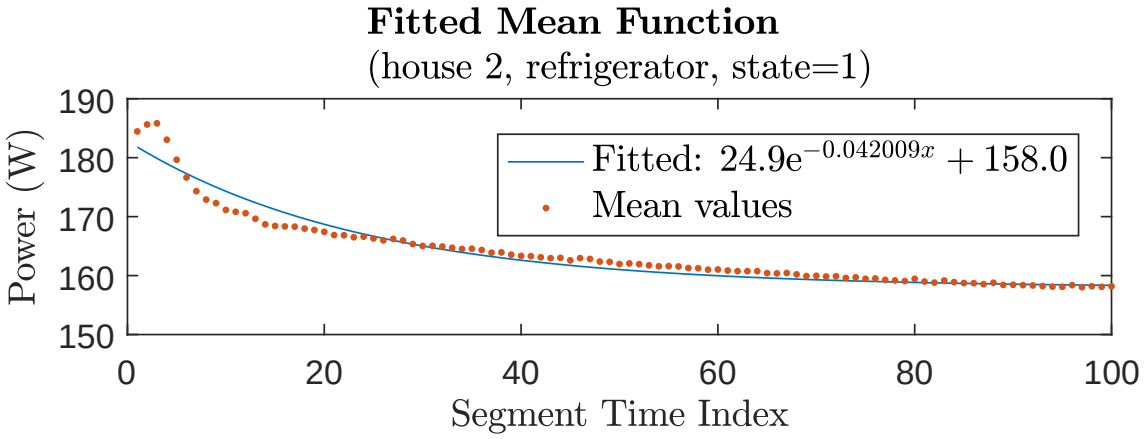



Figure 4.43: Fitted mean function for state 1 of the refrigerator from house 2 of the REDD dataset.

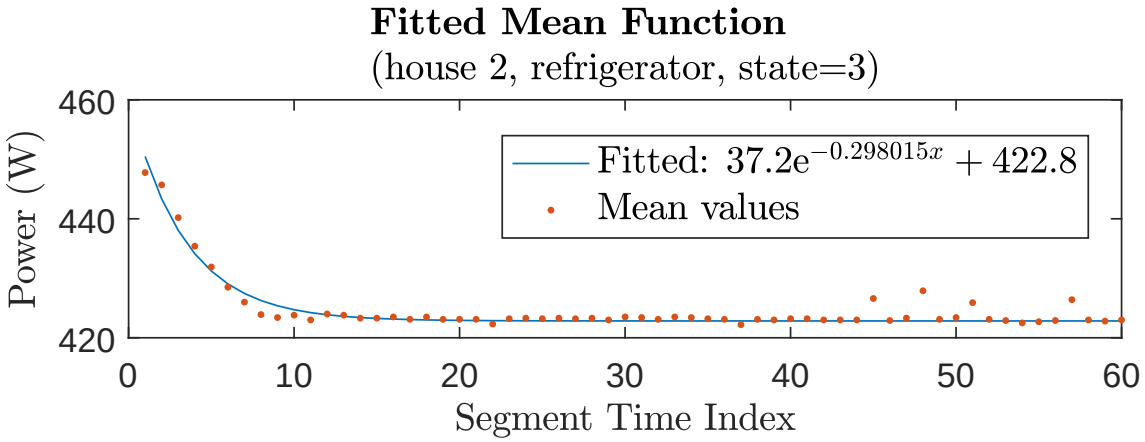



Figure 4.44: Fitted mean function for state 3 of the refrigerator from house 2 of the REDD dataset.

segment time index in the figures is equivalent to the counter values or the dwell time for a particular state.

The segments aligned in this way allow the computation of the mean power for a given counter value. For state 1 and state 3 of the refrigerator, these are presented in Figure 4.41 and Figure 4.42. By performing regression analysis on the preceding outcomes, the fitted mean functions are obtained (see Figure 4.43 and Figure 4.44).

### 4.5.4 State Inference Using PBDT

We now briefly investigate the advantage of using the segmental FVTHMM over the non-segmental version from a load disaggregation perspective. As a proof-of-concept, the data from house 2 of the REDD dataset is chosen for this experiment. Also, the only segmental models are those shown in the previous subsection for state 1 and state 3 of the refrigerator. All the other states of other appliances have the parameter $a_{x_{t,k}}$ set to zero, signifying a constant mean for each segment and each state. Disaggregation is performed like before but now, the recursive expression in (4.25) is used for calculating the particle score in the PBDT algorithm.

A comparison between the segmental version and the ordinary FVTHMM-PBDT method in terms of the CAR metric is shown in Figure 4.45. Expectedly, the explicit modelling of the segmental variation does indeed improve disaggregation accuracy.


Figure 4.45: Comparison between FVTHMM-PBDT and Segmental FVTHMM-PBDT in terms of the overall disaggregation accuracy.

A closer look reveals that nearly all of the misclassifications due to the refrigerator's transient have disappeared. If we consider the ground truth of a particular segment of data shown in Figure 4.46 and the estimates using FVHMM-PBDT

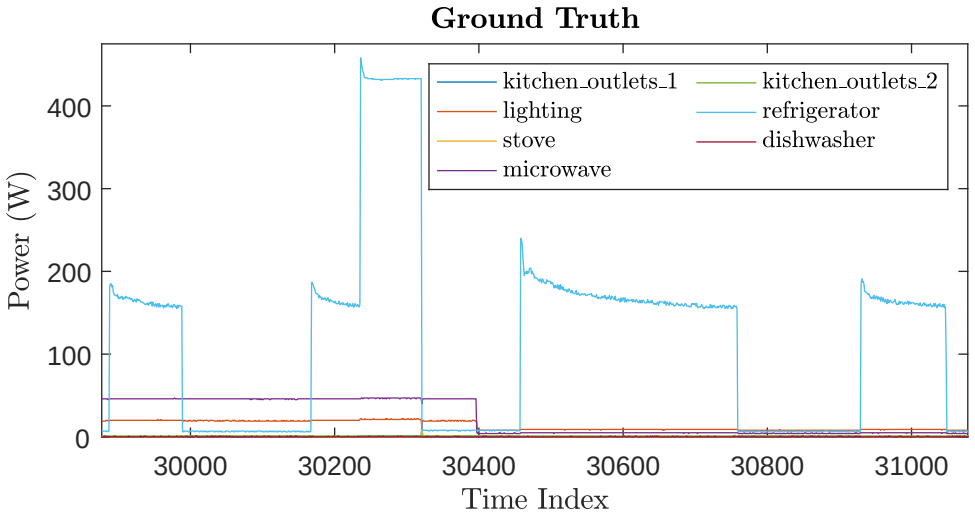



Figure 4.46: Ground truth of a short segment of data from house 2 of the REDD dataset.

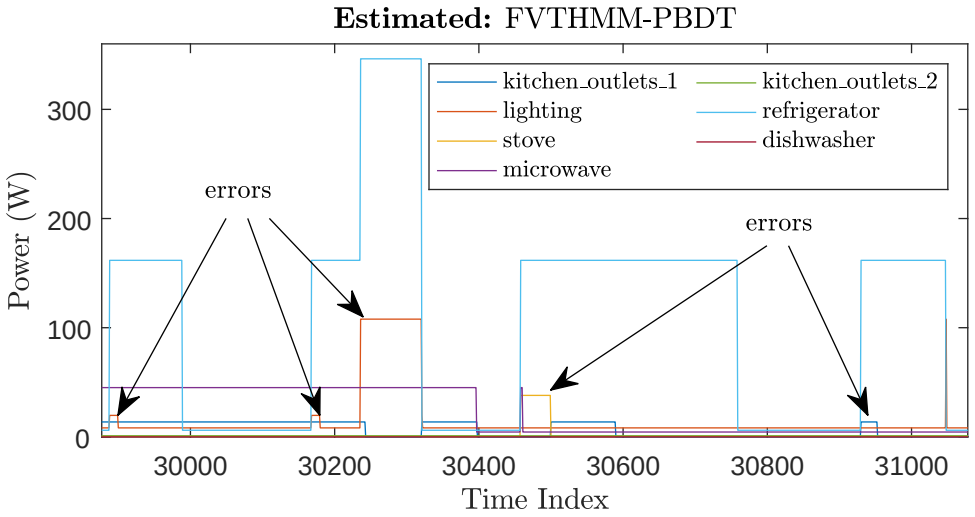



Figure 4.47: The estimates for the same segment using FVTHMM-PBDT.

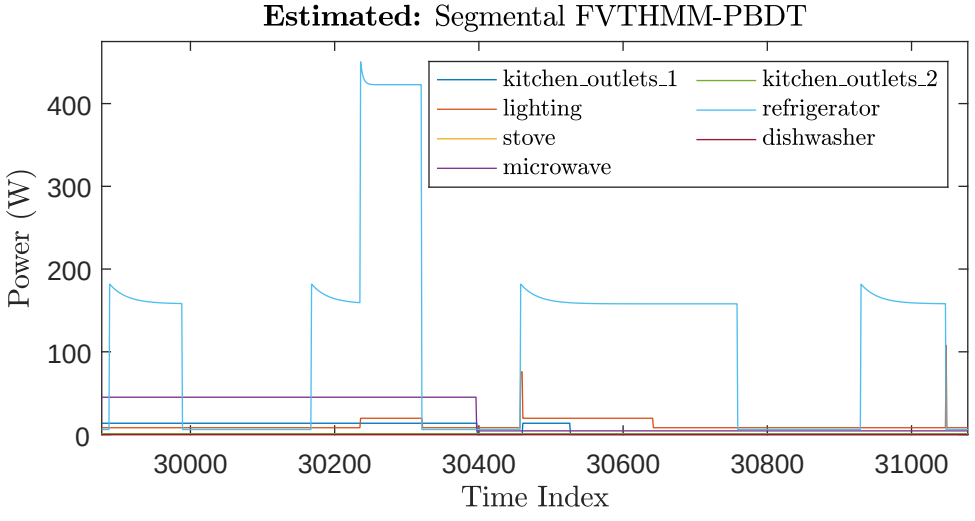



Figure 4.48: The estimates for the same segment using Segmental FVTHMM-PBDT.

shown in Figure 4.47, we can see that there are considerable errors located at the onset of the transients. Specifically, there are the false positives associated with **lighting** and **kitchen_outlets1** at these points.

In contrast, for the segmental counterpart, errors of this kind are largely not present (see Figure 4.48). The only exception is at the time around 30500, where the false positives associated with both **lighting** and **kitchen_outlets1** can be seen. If we turn our attention to the ground truth signal, we can see that this error could very likely be due to the larger-than-usual transient that follows from a possible defrost cycle of the refrigerator at around the time index 30350. The error in particular illustrates why incorporating the variation of variance is important, given that the onset of the transient always have greater variance in general (see the first index of Figure 4.41 and Figure 4.42). However, it is not clear at this point how to best represent the variation of variance in parametric form. It could very well be modelled non-parametrically using Gaussian Processes [Ras04], though these aspects are beyond the scope of this thesis.

## 4.6 Summary

In this chapter, we have presented a new algorithm, Particle-based Distribution Truncation (PBDT), for inferring the state of appliances from the aggregate measurements. Whereas the Viterbi algorithm is computationally intractable to be used with FVTHMM, the proposed algorithm is computationally efficient and scalable. Solutions or particles at each step which are implausible (e.g. low particle score) are truncated in a systematic way or prevented from being generated, mirroring that of the survival-of-the-fittest concept from particle filters, while generated particles with the same augmented system state are merged via a hash-based deduplication procedure to keep only one particle with the highest score, thus closely relating it to the selection step of the Viterbi algorithm in which only the most likely state leading to a given destination state in the trellis structure is kept. In addition, important optimisations included in the implementation of the PBDT algorithm are also described, wherein the distribution of solutions amongst particles at a particular time step is exploited to share computation results for updating the particle score. This was shown to enable an average speed improvement of 20 times over one without such optimisations.

Although the PBDT algorithm is technically an approximation to the Viterbi algorithm, it is inherently a real-time approach, unlike Gibbs sampling and simulated annealing. Further, it has a tunable parameter, i.e. the maximum number of

particles at each time step, to control the extent of the approximation. It was illustrated that, as the parameter increases past a certain point, the PBDT algorithm is able to converge to the optimal state trajectory that would have been produced by the Viterbi algorithm if it were used. As such, the PBDT algorithm is flexible in that it enables the optimality to be controlled, with allowances for finding the optimal solution, should there be computational resources to spare.

The application of the PBDT algorithm to state inferences under FVTHMM for load disaggregation with real-world data reveals a number of important results. Firstly, even when the system state cardinality is in the order of 20 billion, we have demonstrated that the algorithm is able to maintain significant computational throughput (i.e. per-sample processing time of under 1 second) for satisfying real-time requirements while achieving average disaggregation accuracy of approximately 80%. Secondly, the runtime of the algorithm is shown to increase approximately linearly in the number of appliances and approximately logarithmically in the number of system states, validating the claim that the algorithm is scalable.

In evaluating the PBDT algorithm, a new metric for identifying the source of disaggregation errors has also been devised. Known as the cumulative error log-likelihood ratio (CELLR), the metric allows error segments or blocks of consecutive errors to be attributed to the inaccuracies in the model or the approximation inherent to the proposed algorithm. Using this, and its decomposition, $\text{CELLR}_\text{e}$ and $\text{CELLR}_\text{d}$, it was discovered that the majority of errors in the disaggregation of real-world data were not due to the truncation of implausible solutions in the PBDT algorithm but because of spurious observations (e.g. power surge and gradual decays in power) that are difficult to be accounted for in the emission model of FVTHMM.

To overcome this, an augmented form of FVTHMM with segmental emission probabilities has been investigated. The power consumption within a given state is no longer assumed to be stationary. Instead, its distributional parameters are modelled to vary according to the state dwell time, much like the duration-dependent state transition probabilities used in the temporal part of FVTHMM. In the brief evaluation conducted on the real-world data of one house, the augmented model has been shown to fit the gradual decreases in power consumption well, with a further improvement of approximately 5% in the disaggregation accuracy. These preliminary results provide the motivation for more experiments to be done, so that its advantage in modelling a broader range of household appliances can be ascertained. Additionally, the augmented model presented here pro-

vides the framework for which further studies could be performed. Of particular interest is the alternative forms of the segmental emission probability appropriate for the modelling of other classes of appliances. Separate means of improving robustness towards other spurious observations (due to unmodelled appliances) are detailed in the next chapter.

CHAPTER FIVE

# ROBUST EXTRACTION OF APPLIANCE POWER

## 5.1 Introduction and Related Work

The techniques studied in previous chapters have been shown to achieve high disaggregation accuracy while meeting real-time requirements. However, one limiting assumption was made in all of these, that is, all appliances in a household are known, and they are included in the model. In reality, however, unknown appliances are bound to be added via new purchases or guest visits after the initial training stage. If these appliances are not taken into account through some special means, the aggregating effect as a result of these new additions would produce erroneous disaggregation under the previously trained model.

A natural way to solve this is to detect the presence of any new appliances from the aggregate measurements, extract their contributions and learn their model parameters. Though a desired goal from a practical point of view, it is not clear how this could be achieved, given that the mere detection of new appliances requires that the NILM system has certain prior knowledge on the features of unknown loads. Unless these features are particularly distinctive from those of modelled appliances, devising an objective rule for the detection of unknown appliances is immensely challenging. As such, instead of specifically concentrating on the detection of unknown appliances in this way, we approach the problem from a different perspective, whereby the power contributions of modelled appliances are robustly extracted from the aggregate measurements in the presence of unmodelled loads. We believe this is a more sensible approach as more information is readily available from already modelled loads.

The concept was first explored by Kolter and Jaakkola [KJ12], in which they included a robust noise term for absorbing power contributions from unmodelled

devices, while simultaneously allowing those that are modelled to be extracted. In particular, the variation of the noise term is assumed to be piece-wise constant, consistent with the way appliances consume power in general. Therefore, when portions of aggregate power that are not likely to be explained by models of known appliances are observed, they are implicitly assigned to the noise term.

In this chapter, a similar concept is adopted and integrated into the existing FVTHMM-PBDT framework, with the aim of increasing the robustness of the extraction process, while inheriting the advantage of the PBDT algorithm in allowing real-time disaggregation of aggregate measurements and the benefits of using state durations in separating between appliances which are similar, unlike the work of Kolter and Jaakkola [KJ12]. An apparent advantage of this integration is, there is no longer the practical requirement to learn the models of each and every appliance in a residential unit. The learning process can instead be focused on appliances of interest (e.g. refrigerator, heater etc.), leaving other miscellaneous devices to be unmodelled. Examples of the latter are appliances which are too insignificant to be modelled, such as garage door openers, clock radios, cordless telephones and lights in places rarely accessed in the dark (e.g. storeroom and spare room). Additionally, the integrated framework enables the power contributions of an appliance to be extracted and subtracted from the aggregate measurements iteratively; the reduced aggregate measurements get simpler after each round, with the leftover unextracted portion being classed as "unmodelled". Such an iterative NILM approach has been briefly mentioned by Parson et al. [PGWR12] and Wong et al. [WWDc13], but its implementation is beyond the scope of this research. Instead, the means of extraction presented here provides the foundation from which more refined iterative NILM approaches can be developed.

In short, the main contributions discussed in this chapter are

- A robust version of FVTHMM proposed in Chapter 3, robust diff-FVTHMM (RdFVTHMM[1]), which allows the extraction of known appliances to be less affected by the potentially changing composition of unknown loads.
- A modification to the original PBDT algorithm devised in Chapter 4, dPBDT, for the robust real-time tracking of both unmodelled and modelled appliances.

[1]To ease readability, RdFVTHMM can be pronounced as "rhythm".

- A RdFVTHMM-dPBDT framework and a detailed study on its ability to perform disaggregation accurately, even when known appliances and unknown loads have similar power consumption.

## 5.2 Effects of Unknown Appliances

As alluded to at the start of the previous section, the presence of unknown loads negatively affects the detection of known appliances. The aggregate power signal is shifted upwards, biasing the contributions of all the other modelled loads, i.e. in the fundamental equation shown in (3.13) of Chapter 3,

$$y_t = \sum_{k=1}^{K} y_{t,k} + r_t, \tag{5.1}$$

the residual $r_t$ is non-zero. As a result, the emission model governing the pre-existing relationship between the aggregate power consumption $y_t$ and the system state $\mathbf{x}_t$ composed of the states of the $K$ modelled appliances is no longer valid; the assumption that the $K$ appliances in the "knowledge base" are the only possible appliances to be encountered during disaggregation is violated, given that $y_t$ could now be potentially made up of power values external to those included in the model.

As an illustrative example of the negative effects, consider Figure 5.1, where the emission distribution for a given system state $\mathbf{x}_t = \mathbf{i}$ is shown. In the event that $r_t$ is non-zero and $\mathbf{i}$ is the actual system state, the observed $y_t$ is shifted to the tail end of the probability distribution, undesirably reducing the likelihood of $\mathbf{i}$ being the estimate. Therefore, the sequence of system states estimated from

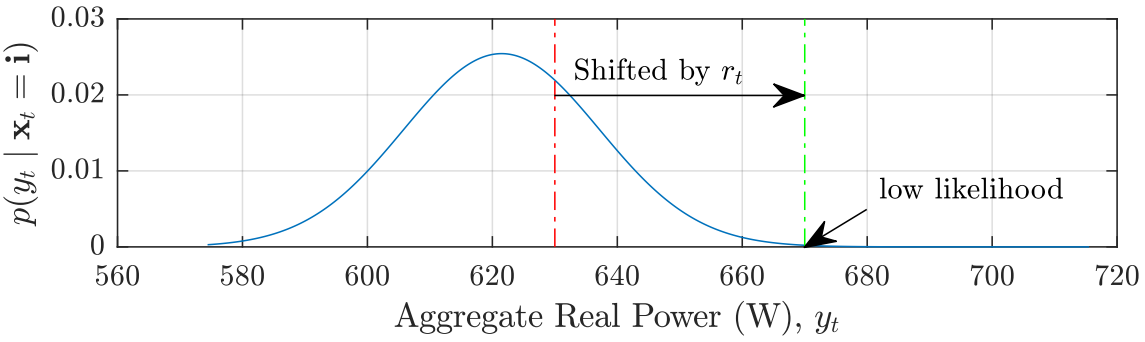
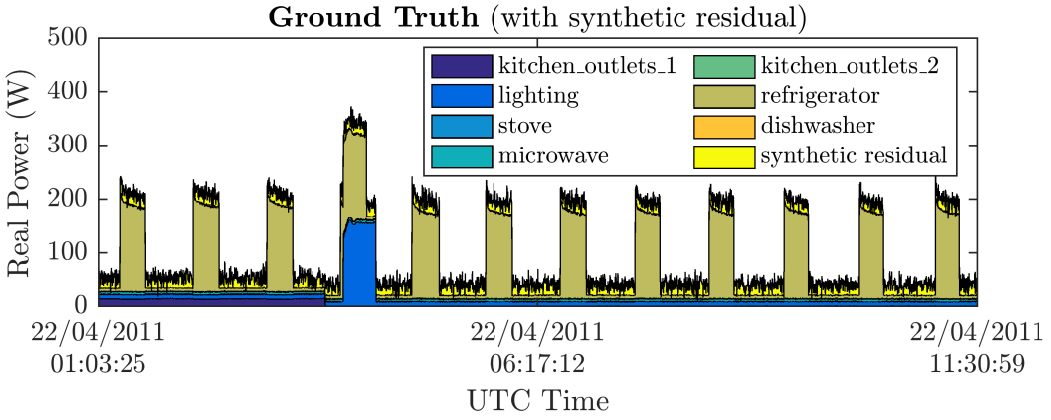
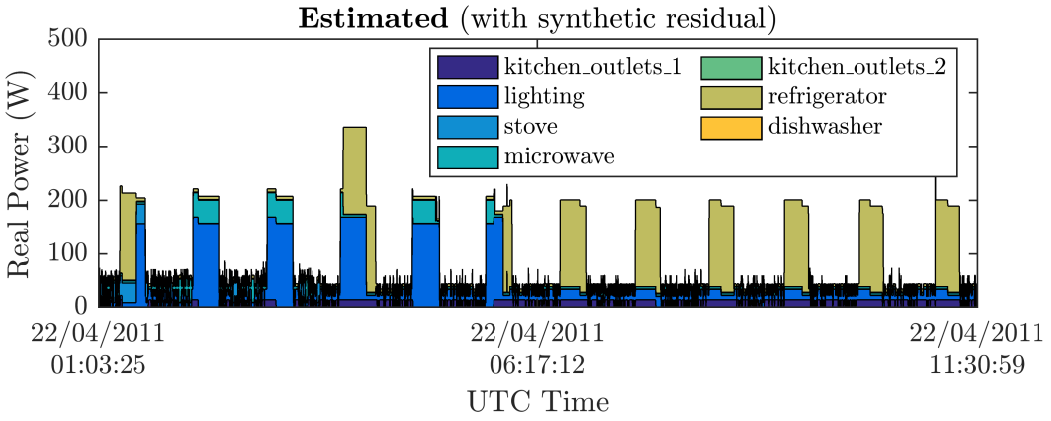


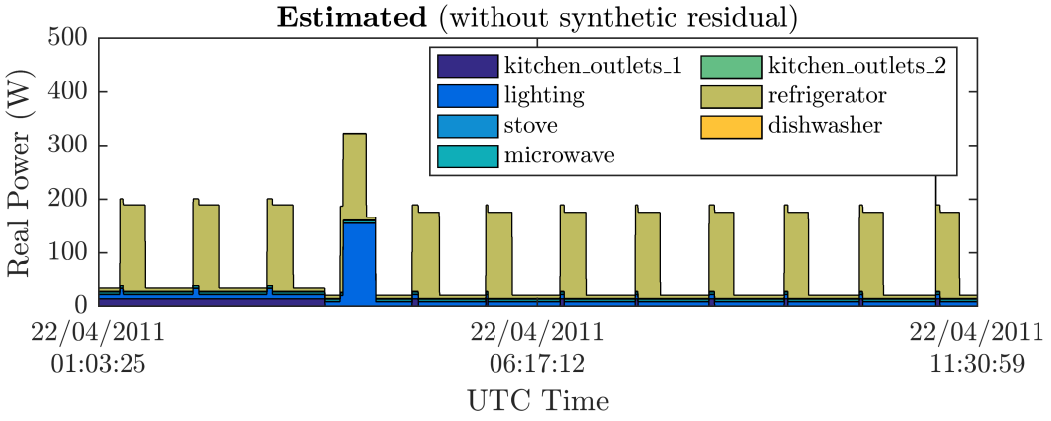

Figure 5.1: The reduced likelihood under a pre-existing model when $r_t$ is non-zero. The red line denotes a particular aggregate power consumption that would have been observed if not for the non-zero $r_t$.




(a) One portion of the aggregate data of house 2 of the REDD dataset with synthetic residual data added.




(b) The estimated contributions of known appliances using FVTHMM-PBDT, when the synthetic residual is present in the aggregate data.




(c) The estimated contributions of known appliances using FVTHMM-PBDT, when the synthetic residual is not present in the aggregate data.

Figure 5.2: A comparison between the output of FVTHMM-PBDT when the synthetic residual is present in the aggregate data and when the synthetic residual is not present in the aggregate data. The synthetic residual shown has a mean of 20 and a standard deviation of 10.

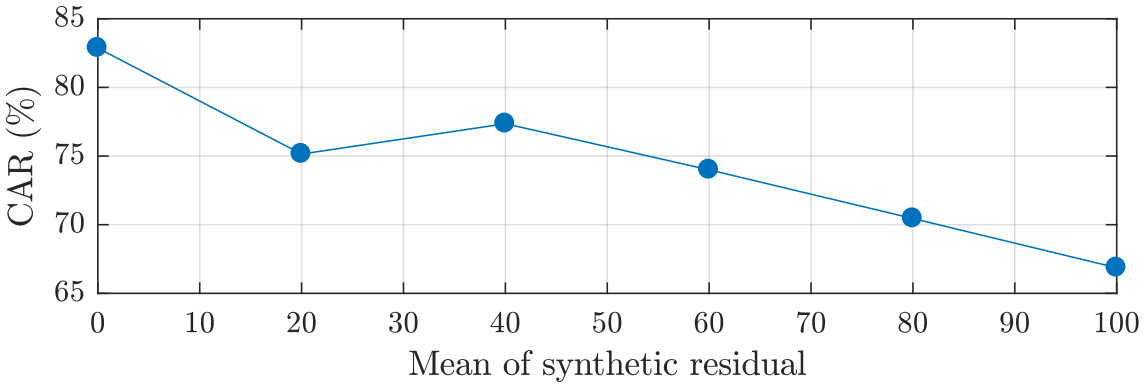



Figure 5.3: The variation of the correct energy assignment rate (CAR) with the mean of the synthetic residual added to the aggregate data.

the maximisation of the overall likelihood $p(\mathbf{x}_{1:T}, y_{1:T}, \mathbf{c}_{1:T})$ could have extreme deviations from the true system states, as compared to when $r_t = 0$ for all $t$.

Figure 5.2 demonstrates a toy example in which data of a synthetic unknown appliance with a mean of 20 and a standard deviation of 10 is added to the aggregate power consumption of house 2 of the REDD dataset. The synthetic data is representative of a small device that is being turned on all the time but unmodelled (e.g. wireless routers, cordless telephones etc.), and the results of the disaggregation show that even with small and relatively constant residual values, errors can be profound. Further, as shown in Figure 5.3, the correct energy assignment rate (CAR) typically reduces with the increasing mean of the synthetic residual.

A natural solution to this problem is to use the change in power consumption, as was done in the seminal paper by Hart [Har92]. This way, constant offsets in the aggregate signal due to unknown loads do not affect the estimation of the states of the known appliances, and observed changes in power which are produced by unknown loads could be ignored if they fall below a certain tolerance level. While the idea was laid out by Hart, it was not specified how such tolerance could be determined so as to account for the variability in the load and other cases where the known appliances are similar in power consumption to unknown loads. Also, besides the limitation in his work where appliances are modelled to have only two states, it is not clear how state duration information, as used in our model, could be seamlessly integrated in his original proposal to aid better separation between similar appliances. For this reason, a more integrated and systematic approach is required.

## 5.3 A Robust Extension of FVTHMM

The previous section has highlighted the practical concerns pertaining to the presence of unknown appliances in a NILM system, and the reduced disaggregation accuracies that result when these appliances are not taken into account. Here, for the robust extraction of the contributions of modelled appliances, we describe a necessary modification to the original FVTHMM. Also discussed is the parameter estimation procedure for the updated model.

### 5.3.1 Model Description

From the issues given in the previous section, it is clear that the modified model has to be able to cope with offsets in power consumption caused by unknown appliances, while not wrongly attributing their power contributions to modelled loads. For this reason, two new additions are incorporated into the original FVTHMM model. The first is the adoption of the change in power as an additional observed variable, like previously mentioned for the work of Hart [Har92], among few others [PGWR12]. This allows the extraction of power consumption of modelled appliances to be more robust against the aforementioned offsets. On the other hand, to prevent changes in power due to unknown appliances from being attributed to any of the modelled devices mistakenly, a noise model inspired by the work of Kolter and Jaakkola [KJ12] is used. The model's role is akin to having an extra component in mixture modelling for capturing outliers, except that, in our case, the outliers refer to power contributions owing to unmodelled loads that are not likely to be generated by the modelled appliances.

Altogether, the outcome of these additions is the robust version of FVTHMM – RdFVTHMM – whose dynamic Bayesian network (DBN) representation is shown in Figure 5.4. In the figure, $z_t$ denotes the difference between the aggregate measurements at time $t$ and time $t-1$ (i.e. $z_t = y_t - y_{t-1}$), whereas $r_t$ refers to the residual at time $t$. The conditional dependence of $z_t$ is expressed by $p(z_t \mid \mathbf{x}_t, \mathbf{x}_{t-1}, r_t, r_{t-1})$, since $z_t$ is mathematically dependent on $y_t$ and $y_{t-1}$. As such, it is by extension dependent on the latent variables, $\mathbf{x}_t$, $\mathbf{x}_{t-1}$, $r_t$, and $r_{t-1}$. Note that, unlike the ordinary FVTHMM mentioned in previous chapters, $\mathbf{x}_t$ now only consists of the states of the $K$ modelled appliances, i.e. $\mathbf{x}_t = (x_{t,1}, \ldots, x_{t,K})$. The contributions of the unknown loads to the aggregate measurements are represented by the residuals. Also, by virtue of the expression in (5.1), the aggregate measurement $y_t$ is now conditionally dependent on both $\mathbf{x}_t$ and the residual, i.e. $p(y_t \mid \mathbf{x}_t, r_t)$.

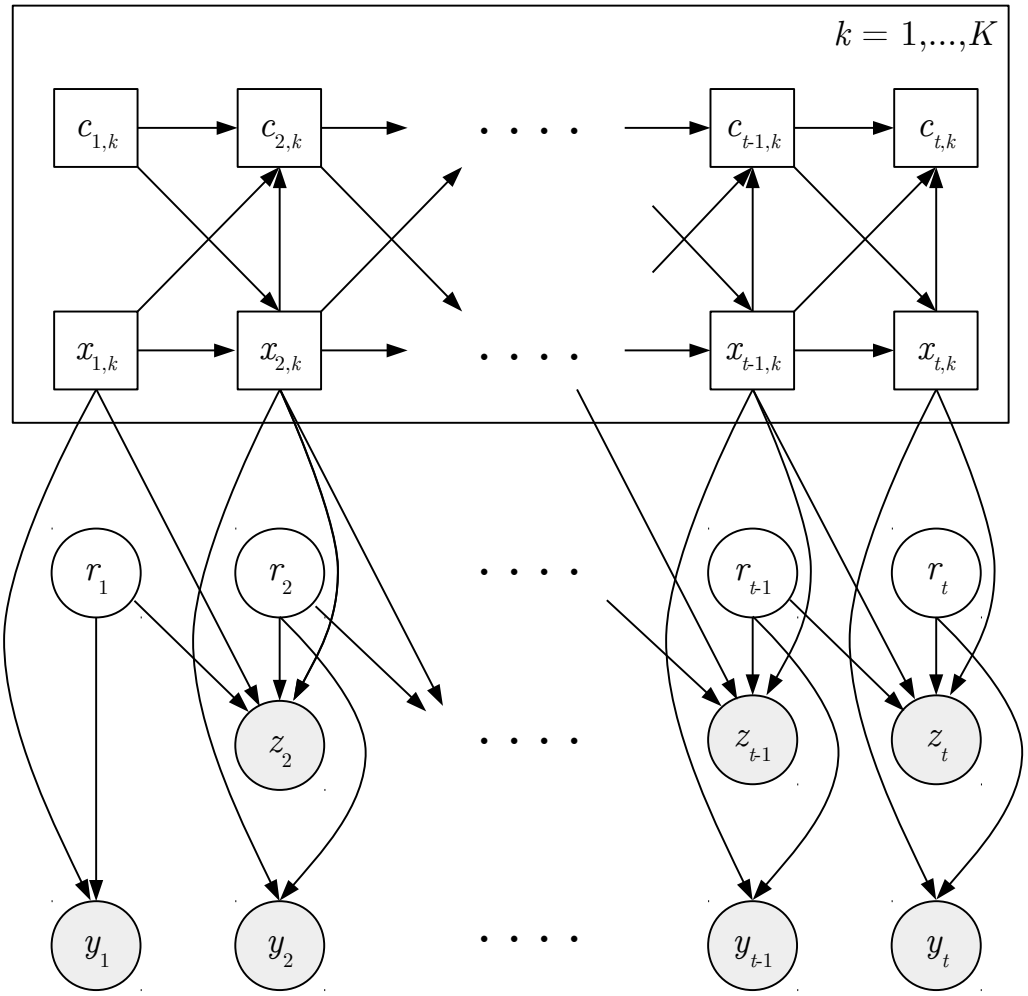



Figure 5.4: Dynamic Bayesian network of the RdFVTHMM.

Given these conditional dependence assumptions, the task is then to specify the form of $p(z_t \mid \mathbf{x}_t, \mathbf{x}_{t-1}, r_t, r_{t-1})$, $p(y_t \mid \mathbf{x}_t, r_t)$ and $p(r_t)$. For the former, we have to consider the variation of $z_t$ with respect to the fluctuations in the aggregate measurements due to modelled appliances, as well as the residuals. When the observed $z_t$ is only caused by the change in states among the modelled loads (i.e. $\mathbf{x}_t \neq \mathbf{x}_{t-1}$ and $r_t = r_{t-1}$), the variation is simply governed by a Gaussian distribution whose mean and variance are those associated with the the transition from $\mathbf{x}_{t-1}$ to $\mathbf{x}_t$; that is, the mean is $\mu_{\mathbf{x}_{t-1},\mathbf{x}_t} = \sum_{k \in H} \mu_{x_{t-1,k}x_{t,k}}$ and the variance is $\sigma^2_{\mathbf{x}_{t-1}\mathbf{x}_t} = \sum_{k \in H} \sigma^2_{x_{t-1,k}x_{t,k}}$, with $H = \{k \in \{1, \cdots, K\} \mid x_{t-1,k} \neq x_{t,k}\}$. $\mu_{x_{t-1,k}x_{t,k}}$ and $\sigma^2_{x_{t-1,k},x_{t,k}}$ denote the mean and variance of the change in power due to the state change of modelled appliance $k$. From $H$, the corresponding mean and variance of self transitions are not included in both $\mu_{\mathbf{x}_{t-1}\mathbf{x}_t}$ and $\sigma_{\mathbf{x}_{t-1}\mathbf{x}_t}$ since only changes in steady-state power are considered, as we shall see in Section 5.4.2.

On the other hand, for cases where the observed $z_t$ is only caused by the change in residuals (i.e. $\mathbf{x}_t = \mathbf{x}_{t-1}$ and $r_t \neq r_{t-1}$), the form of $p(z_t \mid \mathbf{x}_t, \mathbf{x}_{t-1}, r_t, r_{t-1})$ is more difficult to be expressed, as the variation of residuals requires prior knowledge of the unmodelled appliances, which are by definition unknown. However, despite this, we can still make the assumption that the residuals are typically piece-wise constant [KJ12], a property that is consistent with the power draw of many appliances; that is, the changes in the residuals are sparse (i.e. mostly zeros) relative to the length of the observations. It is this sparsity that can be exploited to reconstruct the residual signal, as has been done in the field of compressed sensing. In particular, the reconstructed signal is the solution to the $\ell_1$-minimisation problem, or equivalently, the maximum likelihood problem with the signal values being from a Laplace distribution. Therefore, whenever $\mathbf{x}_t = \mathbf{x}_{t-1}$ and $z_t = r_t - r_{t-1}$, $z_t$ is assumed to be distributed according to a Laplace distribution.

For the work presented here, however, the combined case of $z_t$ being caused by both the state changes among the modelled loads and the changes in the residuals is assumed to be unlikely. Investigation into formulations of $p(z_t \mid \mathbf{x}_t, \mathbf{x}_{t-1}, r_t, r_{t-1})$ to handle deviations from such an assumption is a promising direction for future work.

Taken together, the preceding assumptions made in relation to the variation of $z_t$ give rise to

$$p(z_t \mid \mathbf{x}_t, \mathbf{x}_{t-1}, r_t, r_{t-1}) = \begin{cases} \mathcal{N}\left(z_t \mid \mu_{\mathbf{x}_{t-1}\mathbf{x}_t}, \sigma^2_{\mathbf{x}_{t-1}\mathbf{x}_t}\right), & \text{if } \mathbf{x}_{t-1} \neq \mathbf{x}_t \text{ and } r_t = r_{t-1} \\ \dfrac{\lambda}{2}\exp(-\lambda\|z_t\|_1), & \text{if } \mathbf{x}_t = \mathbf{x}_{t-1} \text{ and } r_t \neq r_{t-1} \\ 0, & \text{otherwise,} \end{cases} \tag{5.2}$$

where $\mathcal{N}(z_t \mid \mu_{\mathbf{x}_{t-1}\mathbf{x}_t}, \sigma^2_{\mathbf{x}_{t-1}\mathbf{x}_t})$ is the probability density function of a Gaussian distribution with mean $\mu_{\mathbf{x}_{t-1}\mathbf{x}_t}$ and variance $\sigma^2_{\mathbf{x}_{t-1}\mathbf{x}_t}$, while $\lambda$ is the rate parameter of the Laplace distribution. This concludes the description for expressing $p(z_t \mid \mathbf{x}_t, \mathbf{x}_{t-1}, r_t, r_{t-1})$.

Let us now turn our attention to the specification of $p(r_t)$ and $p(y_t \mid \mathbf{x}_t, r_t)$, both of which are hard to express, given the unknown nature of the composition of unknown appliances, and the mean and variance of the residual. As such, in place of $p(r_t)$ and $p(y_t \mid \mathbf{x}_t, r_t)$, a penalty function is used to characterise the uncertainty in the variation of $r_t$ and $y_t$. The form of the penalty function is defined

through a few prior knowledge inherent in the relation of (5.1). The first is based on the assumption that a typical household environment would not contain any appliances which could supply power back to the grid, i.e. $r_t \geq 0$ and $y_{t,k} \geq 0$ for all $k$. Secondly, the residuals should be less than or equal to the aggregate values. Together, this culminates into a piece-wise relation

$$f_{\text{penalty}}(r_t \mid \mathbf{x}_t, y_t) = \begin{cases} \dfrac{\mathcal{N}(r_t \mid -3\sigma_{\mathbf{x}_t}, \sigma^2_{\mathbf{x}_t})}{\mathcal{N}(0 \mid 0, \sigma^2_{\mathbf{x}_t})}, & r_t \leq -3\sigma_{\mathbf{x}_t} \\ 1, & -3\sigma_{\mathbf{x}_t} < r_t \leq y_t \\ 0, & r_t > y_t, \end{cases} \tag{5.3}$$

where $\sigma^2_{\mathbf{x}_t} = \sum_{k=1}^{K} \sigma^2_{x_{t,k}}$.

Note that we have chosen to use a smooth transition from the onset of $r_t \leq -3\sigma_{\mathbf{x}_t}$ instead of a sharp fall-off at $r_t = 0$, as for a given $\mathbf{x}_t$, the actual power consumption of appliances are not strictly constant with values equal or more than the mean, $\mu_{\mathbf{x}_t} = \sum_{k=1}^{K} \mu_{x_{t,k}}$; they may have tendencies to fall below the mean as well. Therefore, small amounts of negative $r_t$ have to be tolerated. In the penalty function, the rate of the gradual roll-off towards zero is defined by a renormalised single-sided Gaussian probability density function centred at 3 standard deviations away from the $r_t = 0$, with a variance from the conditional dependence of $y_t$ on $\mathbf{x}_t$ as if $r_t$ were zero, i.e. $p(y_t \mid \mathbf{x}_t)$ in Chapter 3. The penalty function serves to discourage the selection of system states that violate the aforementioned assumptions of $r_t$, so that the estimated $\mathbf{x}_{1:T}$ is consistent with the physical processes of the appliances involved. A visual representation of a penalty function with $y_t = 100$ and $\sigma_{\mathbf{x}_t} = 4$ is shown in Figure 5.5.

By combining the previously mentioned assumptions and the conditional dependence relationships, the DBN shown in Figure 5.4 is thus described by the joint probability

$$\begin{aligned} p(\mathbf{x}_{1:T}, y_{1:T}, z_{2:T}, r_{1:t}, \mathbf{c}_{1:T}) = p(\mathbf{x}_1)p(\mathbf{c}_1) \prod_{t=1}^{T} f_{\text{penalty}}(r_t \mid \mathbf{x}_t, y_t) \\ \times \prod_{s=2}^{T} \Big[ p(\mathbf{x}_s \mid \mathbf{x}_{s-1}, \mathbf{c}_{s-1}) \\ \times p(\mathbf{c}_s \mid \mathbf{x}_s, \mathbf{c}_{s-1}, \mathbf{x}_{s-1}) \\ \times p(z_s \mid \mathbf{x}_s, \mathbf{x}_{s-1}, r_s, r_{s-1}) \Big], \end{aligned} \tag{5.4}$$

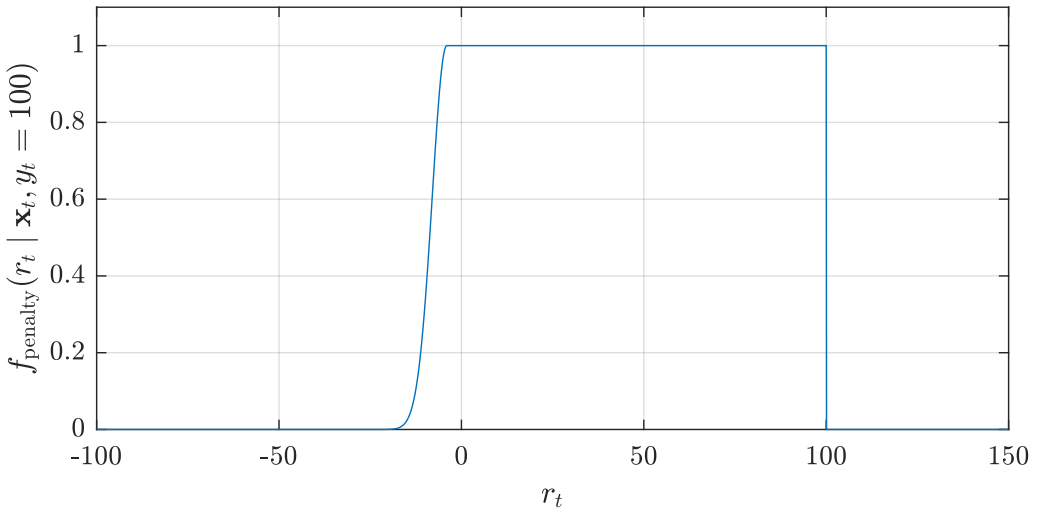



Figure 5.5: Penalty function with $y_t = 100$ and $\sigma_{\mathbf{x}_t} = 4$

while the recursive expression of the same joint probability is

$$\begin{aligned} p(\mathbf{x}_{1:t}, y_{1:t}, z_{2:t}, r_{1:t}, \mathbf{c}_{1:t}) = p(&\mathbf{x}_{1:t-1}, y_{1:t-1}, z_{2:t-1}, r_{1:t-1}, \mathbf{c}_{1:t-1}) \\ &\times f_{\text{penalty}}(r_t \mid \mathbf{x}_t, y_t) \\ &\times p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{c}_{t-1}) \\ &\times p(\mathbf{c}_t \mid \mathbf{x}_t, \mathbf{c}_{t-1}, \mathbf{x}_{t-1}) \\ &\times p(z_t \mid \mathbf{x}_t, \mathbf{x}_{t-1}, r_t, r_{t-1}). \end{aligned} \tag{5.5}$$

### 5.3.2 Parameter Estimation

Before describing the method for estimating the parameters of RdFVTHMM, let us recall from Section 3.4 of Chapter 3 that the model parameters can be denoted by the tuple $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_{\text{e}}, \boldsymbol{\lambda}_{\text{d}})$, where $\boldsymbol{\lambda}_{\text{e}}$ represents the parameters for the emission model, while $\boldsymbol{\lambda}_{\text{d}}$ refers to the parameters for the temporal model.

For RdFVTHMM, $\boldsymbol{\lambda}_{\text{d}}$ is unchanged from that of the FVTHMM since conditional dependencies governing the state transitions and state durations remain the same. Thus, $\boldsymbol{\lambda}_{\text{d}}$ is still composed of the parameters for the mixture of Gamma distribution for each appliance $k$, i.e. $\boldsymbol{m}_k$, the mixture coefficients; $\boldsymbol{\alpha}_k$, the shape parameters; $\boldsymbol{\beta}_k$, the scale parameters. Further, it includes the initial state transition probabilities $\boldsymbol{\pi}_k$ and the Markov state transition matrices $\boldsymbol{A}_k$. In short, $\boldsymbol{\lambda}_{\text{d}} = [(\boldsymbol{\pi}_k, A_k, \boldsymbol{m}_k, \boldsymbol{\alpha}_k, \boldsymbol{\beta}_k)]_{k=1}^{K}$. As these parameters are exactly the same as before, finding their estimates from the training data is not discussed here; their details can be found in Section 3.4.2.

Likewise, $\boldsymbol{\lambda}_{\text{e}}$ is similar to that of the FVTHMM, except it now has to include the parameters of the probability $p(z_t \mid \mathbf{x}_t, \mathbf{x}_{t-1}, r_t, r_{t-1})$; that is, the rate parameter $\lambda$ of the Laplace distribution, and the mean and variance of the change in power as resulting from the transition from state $i$ to $j$ of each known appliance $k$, i.e. $\mu_{i,j}$ and $\sigma^2_{i,j}$.

### Finding the rate parameter

To determine $\lambda$, a natural approach is to solve

$$\hat{\lambda} = \arg\max_{\lambda} p(\mathbf{x}_{1:T}, y_{1:T}, z_{1:T}, r_{1:T}, \mathbf{c}_{1:T}) \tag{5.6}$$

over a training dataset with known $\mathbf{x}_{1:T}$, $y_{1:T}$, $z_{1:T}$, $r_{1:T}$ and $\mathbf{c}_{1:T}$. Because $p(z_t \mid \mathbf{x}_t, \mathbf{x}_{t-1}, r_t, r_{t-1})$ is the only factor that depends on $\lambda$, and as the Laplace distribution only arises for cases with $\mathbf{x}_{t-1} = \mathbf{x}_t$ and $r_t \neq r_{t-1}$, the problem can be rewritten as

$$\hat{\lambda} = \arg\max_{\lambda} \sum_{s \in \mathcal{Z}} \log(p(z_s \mid \mathbf{x}_s, \mathbf{x}_{s-1}, r_s, r_{s-1})), \tag{5.7}$$

where $\mathcal{Z} = \{t \mid \mathbf{x}_{t-1} = \mathbf{x}_t\}$. After performing the relevant derivations, the solution reduces to a closed form, i.e.

$$\hat{\lambda} = \frac{|\mathcal{Z}|}{\sum_{s \in \mathcal{Z}} \|z_s\|_1}, \tag{5.8}$$

with $|\mathcal{Z}|$ denoting the cardinality of the set $\mathcal{Z}$. Not surprisingly, this is the same expression as the maximum-likelihood estimate (MLE) of $\lambda$ for a typical Laplace distribution. In this context however, it can be interpreted as the reciprocal of the average change in power attributed to unknown loads in the training data. If the change in power, when none of the known appliances change states, is large on average, $\hat{\lambda}$ will be low and the standard deviation of the fitted Laplace distribution will be high to accommodate changes in the residuals which deviate further away from zero.

Although the use of (5.8) in this way appears to be an elegant approach, the MLE of $\lambda$ is not an obvious right choice, given the potential differences in what constitutes an unmodelled load in the training data and those encountered during disaggregation after deployment. While this may suggest that determining $\lambda$ is difficult, a guideline for tweaking $\lambda$ can actually be made. In particular, we can consider the role of $\lambda$ in influencing the height and spread of the Laplace probability density function (pdf) relative to those of the Gaussian pdfs for cases

where $\mathbf{x}_{t-1} \neq \mathbf{x}_t$ and $r_t = r_{t-1}$. This can be defined by a flatness ratio $\rho_\sigma$ between the standard deviation of the Laplace pdf, $\sigma_\lambda = \sqrt{2}/\lambda$, and the largest standard deviation among the Gaussian pdfs for the state transitions, i.e.

$$\rho_\sigma = \frac{\sigma_\lambda}{\max\limits_{k} \max\limits_{\substack{x_{t-1,k}, x_{t,k} \\ x_{t-1,k} \neq x_{t,k}}} \sigma_{x_{t-1,k} x_{t,k}}}. \tag{5.9}$$

Ideally, the Laplace pdf should be much flatter at points where the mass of the Gaussian pdfs are significant, so that change in power values close to the centroid of the Gaussian pdfs are correctly assigned to modelled appliances instead of the residuals. However, there is an inherent trade-off in the choice of $\lambda$ (or equivalently, $\rho_\sigma$) for meeting this goal. To gain an intuition on the role of $\lambda$ in this regard, consider Figure 5.6 and Figure 5.7. The Gaussian pdfs shown are associated with the case for when there is a state transition among the modelled appliances, whereas the Laplace pdfs correspond to the case with no modelled appliances changing states. In the figures, $V$ denotes the difference in the likelihood value between the Gaussian pdf and the Laplace pdf evaluated at $\mu_{\mathbf{x}_{t-1}, \mathbf{x}_t}$,

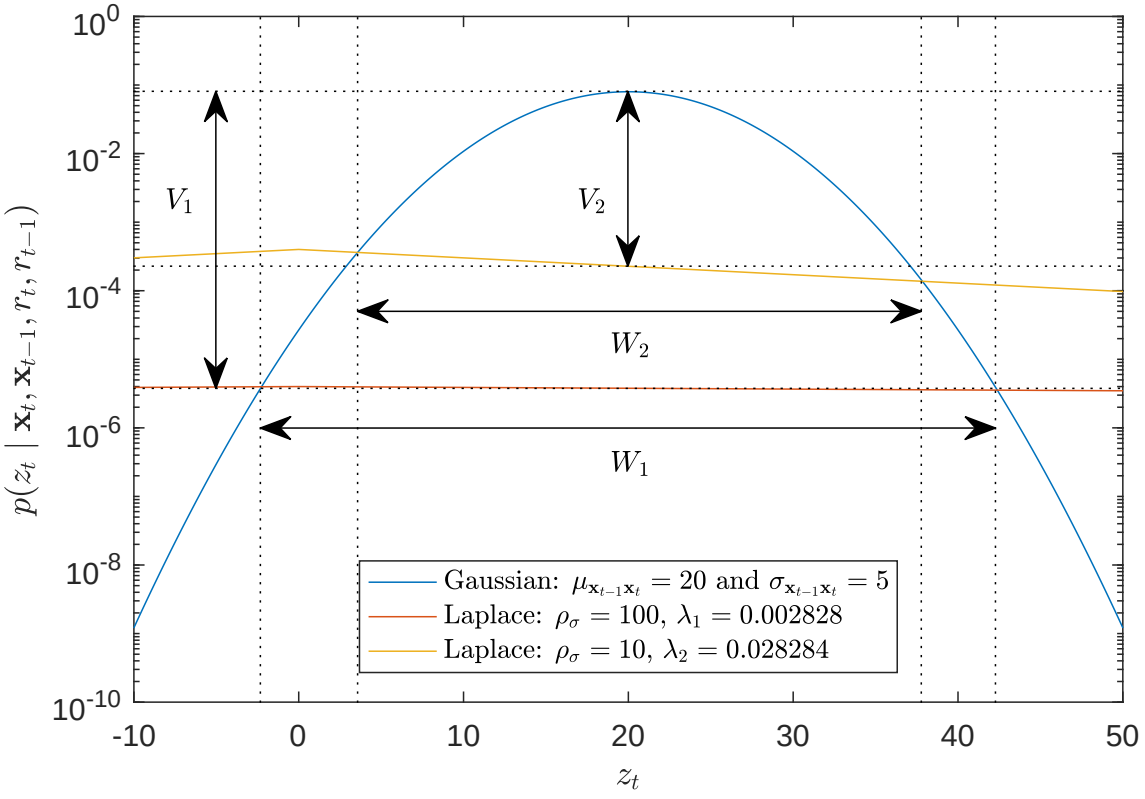



Figure 5.6: Visual representation of (5.2) with small $\mu_{\mathbf{x}-\mathbf{1}_t \mathbf{x}_t}$. The Gaussian distribution corresponds to the case of $\mathbf{x}_{t-1} \neq \mathbf{x}_t$, while the Laplace distributions correspond to the case with no state transitions, each with different $\rho_\sigma$.

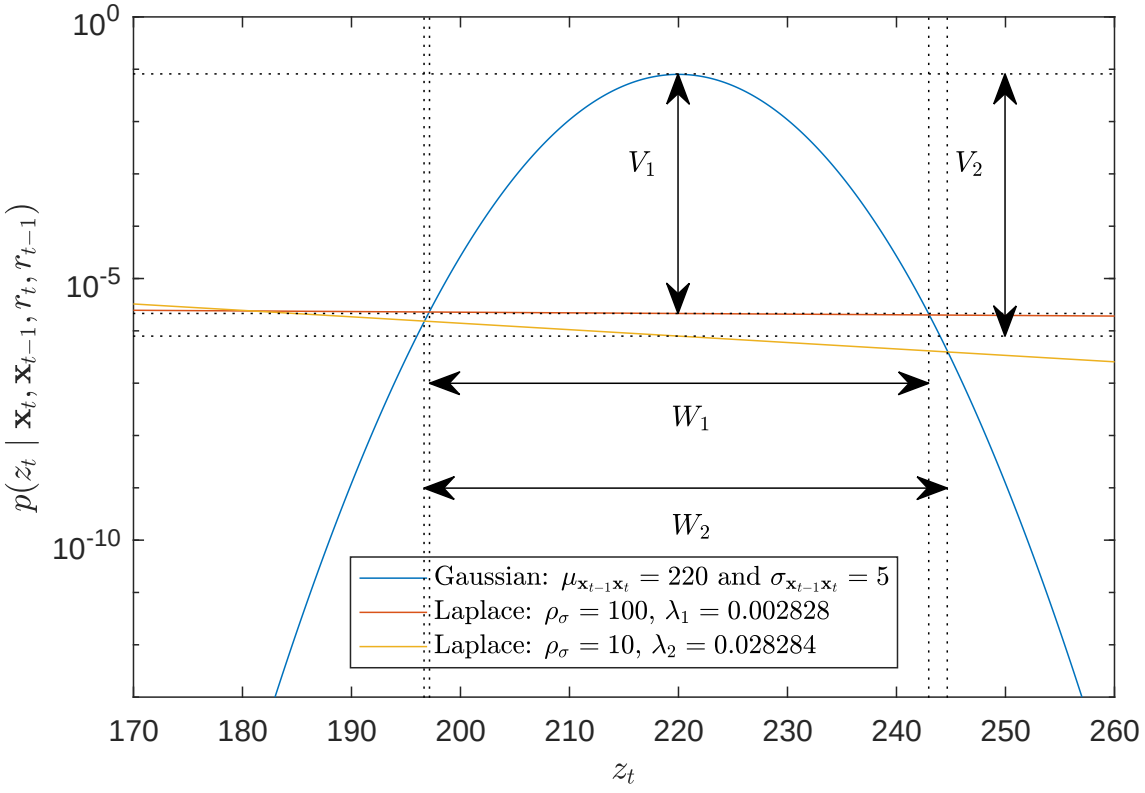



Figure 5.7: Visual representation of (5.2) with large $\mu_{\mathbf{x}-\mathbf{1}_t\mathbf{x}_t}$. The Gaussian distribution corresponds to the case of $\mathbf{x}_{t-1} \neq \mathbf{x}_t$, while the Laplace distributions correspond to the case with no state transitions, each with different $\rho_\sigma$.

while $W$ refers to the range of values for which the likelihood value of the Gaussian pdf exceeds the likelihood of the Laplace pdf.

As illustrated in Figure 5.6, it can be seen that choosing a small $\lambda$ (or large $\rho_\sigma$) leads to a bigger $V$ for state transitions with small $\mu_{\mathbf{x}_{t-1},\mathbf{x}_t}$. On the other hand, as shown in Figure 5.7, selecting a small $\lambda$ (or large $\rho_\sigma$) for cases with big $\mu_{\mathbf{x}_{t-1},\mathbf{x}_t}$ results in small $V$, given the steeper decay of the Laplace pdf with larger $\lambda$. However, increasing $V$ has the side effect of increasing $W$. With a larger $W$, there will be more tendencies for $z_t$ to be assigned to any of the modelled appliances even though the observed $z_t$ is actually caused by unknown loads. As such, there is a trade-off in the choice of $\lambda$ (or $\rho_\sigma$) between selecting for a large $V$ and selecting for a small $V$. In practice, an intermediate $\lambda$ (or $\rho_\sigma$) which balances between the two extremes should be chosen. A more detailed discussion of $\rho_\sigma$ and its effects on the extraction accuracy of known appliances is given in Section 5.5.3.

**Finding the means and variances of known appliances' change in power**

The mean and the variance of the change in power that results from the transition from state $i$ to a different state $j$ (i.e. $i \neq j$) can be derived from the mean and variance of the power consumption of state $i$ and state $j$, learned as part of the training procedure described in Section 3.4.1, such that $\mu_{i,j} = \mu_j - \mu_i$ and $\sigma^2_{i,j} = \sigma^2_i + \sigma^2_j$. On the other hand, for the case of self-transitions (i.e. $i = j$), $\mu_{i,j}$ and $\sigma^2_{i,j}$ are not required to be specified explicitly, since the calculations of $\mu_{\mathbf{x}_{t-1},\mathbf{x}_t}$ and $\sigma_{\mathbf{x}_{t-1},\mathbf{x}_t}$ only perform summations over the parameters corresponding to those appliances which change states, i.e. $\mu_{\mathbf{x}_{t-1}\mathbf{x}_t} = \sum_{k\in H} \mu_{x_{t-1,k}x_{t,k}}$ and $\sigma^2_{\mathbf{x}_{t-1}\mathbf{x}_t} = \sum_{k\in H} \sigma^2_{x_{t-1,k}x_{t,k}}$, with $H = \{k \in \{1,\cdots,K\} \mid x_{t-1,k} \neq x_{t,k}\}$. This is justified by the use of a steady-state segmentation algorithm during disaggregation, where the mean of consecutive power values considered to be in steady-state is subtracted from the mean of the following steady-state segment, as we shall see in the next section.

## 5.4 A Modified PBDT Algorithm

In this section, we first explain the problem of applying the original PBDT algorithm directly to RdFVTHMM. Then, with consideration of these issues, we outline a series of modifications that are needed as part of a modified version of the PBDT algorithm.

### 5.4.1 Overview

In Chapter 4, the original PBDT algorithm has been demonstrated to work particularly well for state inferences under FVTHMM. For the case of RdFVTHMM however, due to the weak constraints imposed by the penalty function and the limited information contained in the change in power signal when no state change occurs, it was found that the algorithm is now more sensitive to transients (e.g. power surge, slow rise-time) that happens at the onset of the state transition of some appliances. In particular, the particle with the true system state (i.e. the true particle) at a certain time step but with an intermediate score as a result of one such observation, now has a reduced ability to gain rank as more subsequent stable values are observed. Worse, it may even lose rank at each increasing time step after observing an atypical change in power, with the potential of being truncated in the end.

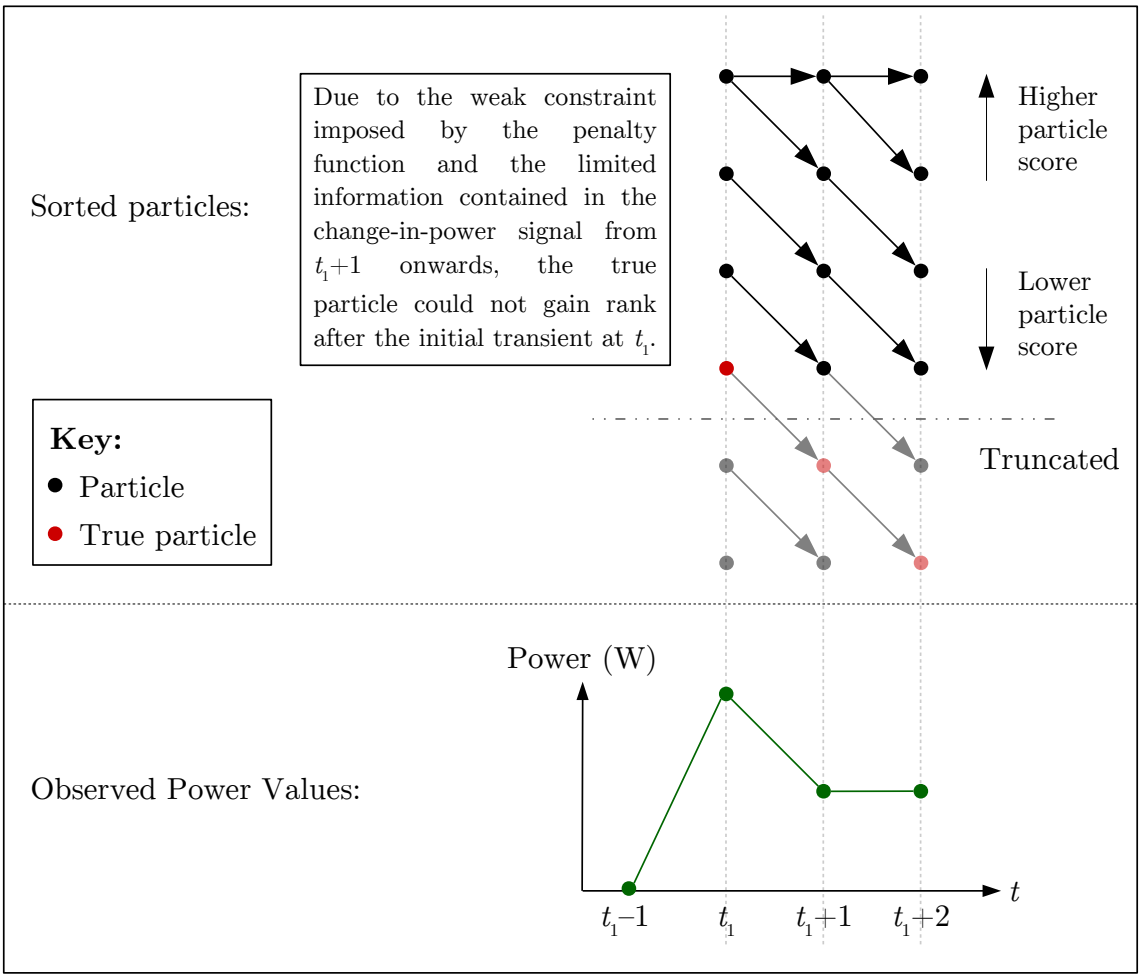



Figure 5.8: Sensitivity to transients.

To better understand this phenomenon, consider Figure 5.8 where a hypothetical aggregate power signal and a sorted list of particles at each time step are shown. At time $t_1$, an appliance is switched on and the power signal increases past its steady-state value. As this surge in power is an atypical event, and it is not accounted for in the model, the score of the true particle is computed to be lower than if a power surge is not observed. Hence, at the expense of the true particle, the other particles with the wrong states which could better explain the anomaly are ranked higher.

Typically, the PBDT algorithm allows the true particle to increase its score if subsequent values after $t_1$ are stable and close to its modelled values. However, because the penalty function $f_{\text{penalty}}(r_t \mid \mathbf{x}_t, y_t)$ is less restrictive in the sense that it is significant over a wider range of values in its domain as compared that of the emission probability $p(y_t \mid \mathbf{x})$ used in FVTHMM, particles at time $t > t_1$, with ancestors at $t_1$ ranked higher than the true particle, are not forced to take on lower

scores. While one might expect the factor $p(z_t \mid \mathbf{x}_t, \mathbf{x}_{t-1}, r_t, r_{t-1})$ to compensate for the less restrictive penalty function, it does not provide enough information for corrective actions to take place, given that the change in power at steady state is normally zero on average, regardless of the appliances that are in operation. Therefore, modifications to the original PBDT algorithm are clearly needed for the state inference under RdFVTHMM. Figure 5.9 illustrates the key components of the modified PBDT algorithm detailed in the discussion that follows.

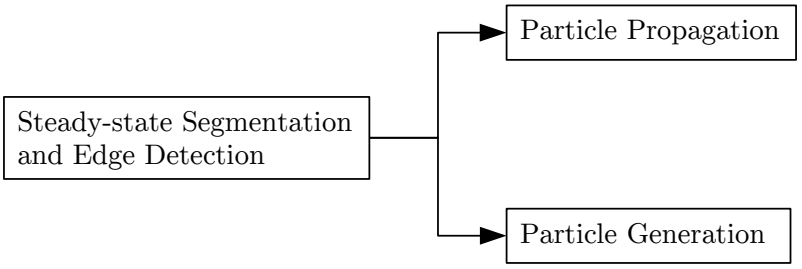



Figure 5.9: The block diagram of the modified PBDT algorithm.

### 5.4.2 Steady-state Segmentation and Edge Detection

The first modification involves the detection of state changes (or edges) from the aggregate power measurements and the extraction of the mean of a segment of consecutive power values considered to be in steady-state. Like in [Har85], the main idea is to use the changes in the mean between two steady-state segments, $\Delta_{\text{ss}} = \mathcal{Y}_{\text{new}} - \mathcal{Y}_{\text{old}}$ (see Figure 5.10), instead of the pair-wise change in power, $z_t = y_t - y_{t-1}$, in the calculation of the particle score. This prevents a significant reduction in the score of the true particle as a result of transient power values.

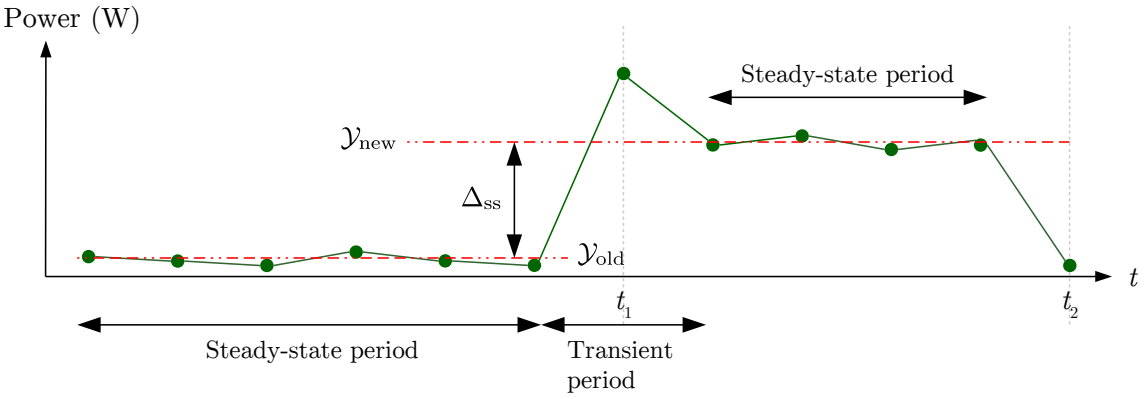



Figure 5.10: The mean of two steady-state segments, $\mathcal{Y}_{\text{old}}$ and $\mathcal{Y}_{\text{new}}$, and the difference between the means, $\Delta_{\text{ss}}$.

Before describing the steady-state segmentation and the edge detection procedure, the notion of steady-state and what constitutes a state change need to be quantified. The condition for the former is said to be met if $N_{\text{ss}}$ consecutive power measurements all have values of $|z_t|$ falling below a certain predefined threshold $\Delta_{\text{thres}}$, while the latter is the reverse case where $|z_t|$ exceeds $\Delta_{\text{thres}}$. We follow Hart's original implementation [Har85] and use a $N_{\text{ss}}$ of 3. On the other hand, the choice of $\Delta_{\text{thres}}$ and its effect on the overall disaggregation accuracy are studied in Section 5.5.3. Admittedly, a more sophisticated version of the procedure could be adopted from the field of change-point detection [AM07]. However, such an implementation is beyond the scope of the research.

The method, based on Hart's work [Har85], for the real-time detection of state changes and the segmentation of steady-state, is summarised in Algorithm 2. If we assume that the mean of a previous steady-state segment, $\mathcal{Y}_{\text{old}}$, has been determined, the power values for the current steady-state segment over the last $\mathcal{C}_{\text{steady}}$ time steps have been stable and the current power measurement $y_t$ does not exceed that of the previous time step by $\Delta_{\text{thres}}$, then the mean for the current steady-state segment up to the current time step $t$ will be updated according to the running average, $(\mathcal{C}_{\text{steady}} \times \mathcal{Y}_{\text{new}} + y_t)/(\mathcal{C}_{\text{steady}} + 1)$. On the other hand, if $y_t$ deviates by $\Delta_{\text{thres}}$ or more from $y_{t-1}$, with other conditions being the same, the current steady-state segment is considered to have ended and therefore, $\Delta_{\text{ss}}$ can be computed. Accordingly, $\mathcal{C}_{\text{steady}}$ is reset to zero so that $\mathcal{Y}_{\text{new}}$ now starts afresh

**Algorithm 2** Real-time Steady-state Segmentation and Edge Detection

**for** each time step $t$ **do**
  **if** $|y_t - y_{t-1}| \leq \Delta_{\text{thres}}$ **then**
    exceeded $\leftarrow 0$
  **else**
    exceeded $\leftarrow 1$
  **end if**
  **if** exceeded = 1 and changing = 0 **then**
    $\Delta_{\text{ss}} \leftarrow \mathcal{Y}_{\text{new}} - \mathcal{Y}_{\text{old}}$
    $\mathcal{Y}_{\text{old}} \leftarrow \mathcal{Y}_{\text{new}}$
  **end if**
  **if** exceeded = 1 **then**
    $\mathcal{C}_{\text{steady}} \leftarrow 0$
  **end if**
  $\mathcal{Y}_{\text{new}} \leftarrow (\mathcal{C}_{\text{steady}} \times \mathcal{Y}_{\text{new}} + y_t)/(\mathcal{C}_{\text{steady}} + 1)$
  $\mathcal{C}_{\text{steady}} \leftarrow \mathcal{C}_{\text{steady}} + 1$
  changing $\leftarrow$ exceeded
**end for**

to accumulate the subsequent observed power measurements towards the mean for the next steady-state segment. Note that the accumulation will only begin once the steady-state condition has been established; power measurements due to transient behaviours are not included in the average.

### 5.4.3 Particle Generation and Propagation

The second modification involves not generating particles at each time step, unlike the original PBDT algorithm described in Section 4.3.1 of Chapter 4. Instead, particles are only generated at points in time where state change occurs or when change in steady-state power consumption is detected (i.e. at only event points). Between two event points $t_1$ and $t_2$, with $t_1 < t_2$, the particles are simply propagated from $t_1$ to $t_2 - 1$. That is, their system states within the interval are fixed, but their counter vectors $\mathbf{c}_{t_1:t_2-1}$ and their scores are updated accordingly. Not only does this prevent erroneous generation of sorted particles at each time step when information that can be gleaned from the change in power values is limited, it also nicely integrates with the steady-state segmentation and the edge detection procedures described in Section 5.4.2.

The overall process is presented in Figure 5.11. At time $t_1$ or whenever $\Delta_{\text{ss}}$ has been calculated, the particle generation procedure detailed in Section 4.3.1 is performed to create $N_{\text{p,max}}$ particles. More specifically, for reasons already explained in the previous chapter and to ensure the efficient enumeration of the possible $\mathbf{x}_{t_1}$ when generating the offspring particles for each $m$th parent particle, only the system states $\mathbf{x}_{t_1}$ satisfying $f_{\text{penalty}}(r_{t_1} \mid \mathbf{x}_{t_1}, y_{t_1}) > \epsilon$ and $p(\Delta_{\text{ss}} \mid \mathbf{x}_{t_1}, \hat{\mathbf{x}}_{t_1-1}^{(m)}, r_{t_1}, \hat{r}_{t_1-1}^{(m)}) > \epsilon$, and those corresponding to at most three appliances changing states (i.e. the Hamming distance $d_H(\hat{\mathbf{x}}_{t_1-1}^{(m)}, \mathbf{x}_{t_1}) \leq 3$) are considered. Then, based on the recursive expression of the joint probability of RdFVTHMM in (5.5), each $n$th generated particle at $t = t_1$ is scored using

$$\mathcal{S}_t(n) = \begin{cases} \log(p(\hat{\mathbf{x}}_1^{(n)}, \hat{\mathbf{c}}_1^{(n)})) + \log(f_{\text{penalty}}(\hat{r}_1^{(n)} \mid \hat{\mathbf{x}}_1^{(n)}, y_1)), & \text{if } t = 1 \\ \mathcal{S}_{t-1}(m) + \log(f_{\text{penalty}}(\hat{r}_t^{(n)} \mid \hat{\mathbf{x}}_t^{(n)}, y_t)) & \\ + \log(p(\hat{\mathbf{x}}_t^{(n)}, \hat{\mathbf{c}}_t^{(n)} \mid \hat{\mathbf{x}}_{t-1}^{(m)}, \hat{\mathbf{c}}_{t-1}^{(m)})) & , \quad \text{if } t > 1, \\ + \log(p(\Delta_{\text{ss}} \mid \hat{\mathbf{x}}_t^{(n)}, \hat{\mathbf{x}}_{t-1}^{(m)}, \hat{r}_t^{(n)}, \hat{r}_{t-1}^{(m)})) & \end{cases} \tag{5.10}$$

before being sorted and truncated to keep only the $N_{\text{p,max}}$ particles with the highest score, like in the original PBDT algorithm. Having generated the particles at

$t_1$, the propagation step is trivial; the particles for $t \in [t_1 + 1, t_2 - 1]$ are forced to have their ranks and system states unchanged (i.e. $n = m$ and $\hat{\mathbf{x}}_t^{(n)} = \hat{\mathbf{x}}_{t-1}^{(m)}$), whereas $\hat{\mathbf{c}}_t = \hat{\mathbf{c}}_{t-1}^{(n)} + 1$ and the scores are updated as in (5.10) but with $\Delta_{\text{ss}} = 0$.

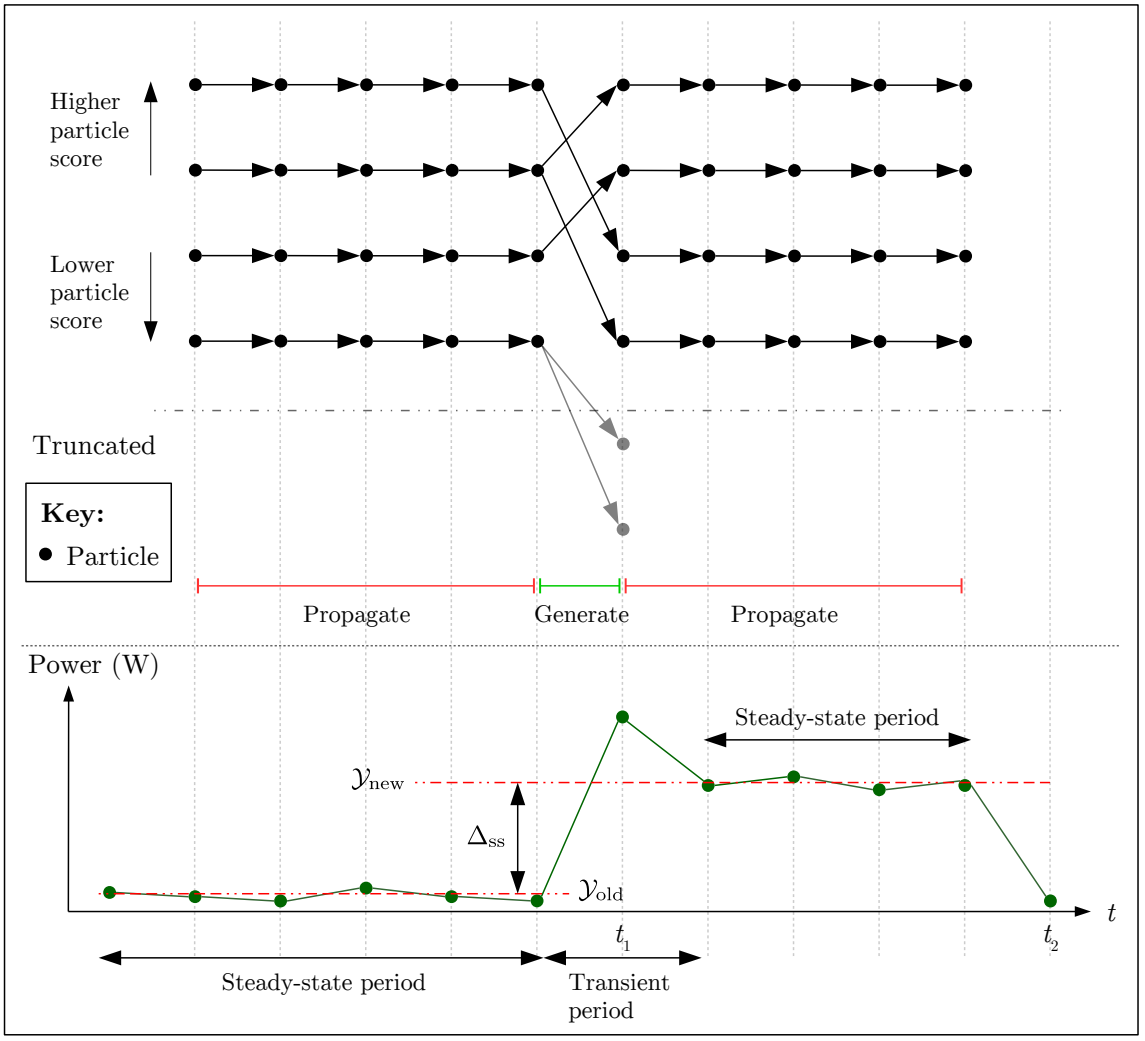



Figure 5.11: The process of particle generation and propagation in the modified PBDT algorithm, dPBDT.

In the implementation, further computational optimisations based on the sharing of computation results discussed in Section 4.3.2 are included. Rather than calculating $p(\Delta_{\text{ss}} \mid \mathbf{x}_t, \hat{\mathbf{x}}_{t-1}^{(m)}, \hat{r}_t^{(n)}, \hat{r}_{t-1}^{(m)})$ for each $m$th parent particle, we exploit the observation that $\mathbf{x}_{t-1}$ for different parent particles can be common. In this way, calculations of $p(\Delta_{\text{ss}} \mid \mathbf{x}_t, \hat{\mathbf{x}}_{t-1}^{(m)}, \hat{r}_t^{(n)}, \hat{r}_{t-1}^{(m)})$ for a given group of parent particles with the same $x_{t-1}$ can be shared, and computational improvements could be gained. To recall more details on this computation-sharing scheme, see Section 4.3.2.

## 5.5 Experimental Results and Discussion

In this section, we evaluate the disaggregation accuracy of the modified method, RdFVTHMM-dPBDT, for cases where not all appliances in a residential unit are modelled or known. Using both synthetic data and data of real homes from the REDD dataset, comparison of the method is made with respect to FHMM-PBDT, FVTHMM-PBDT and the robust version of FHMM-PBDT, RdFHMM-dPBDT. Also presented is an empirical study of the influence of the flatness ratio $\rho_\sigma$ and the threshold $\Delta_{\text{thres}}$ on the disaggregation accuracy. Like in Chapter 4, all evaluations are performed using MATLAB on a PC with an Intel Core i7-4770 processor and 16 GB of RAM.

### 5.5.1 Evaluation Metrics

To quantify the correct extraction of known appliances in the presence of unknown loads, three additional metrics are used, in addition to those introduced in Section 4.4.1, since the previous metrics do not provide sufficient insights necessary for understanding wrong energy assignments to unmodelled and modelled appliances. Basing off the standard precision, recall and F-score metrics from the field of information retrieval [KKP06], they are

$$\mathscr{P}_k = \frac{\mathrm{E}_{\mathrm{TP},k}}{\mathrm{E}_{\mathrm{TP},k} + \mathrm{E}_{\mathrm{FP},k}} \tag{5.11}$$

$$\mathscr{R}_k = \frac{\mathrm{E}_{\mathrm{TP},k}}{\mathrm{E}_{\mathrm{TP},k} + \mathrm{E}_{\mathrm{FN},k}} \tag{5.12}$$

$$\mathscr{F}_k = 2 \cdot \frac{\mathscr{P}_k \mathscr{R}_k}{\mathscr{P}_k + \mathscr{R}_k}. \tag{5.13}$$

$\mathrm{E}_{\mathrm{TP},k}$, $\mathrm{E}_{\mathrm{FN},k}$ and $\mathrm{E}_{\mathrm{FP},k}$ are defined as before in Section 4.4.1, while $\mathscr{P}_k$ and $\mathscr{R}_k$ are the energy-assignment variant of the precision and recall of appliance $k$ respectively. The F-score $\mathscr{F}_k$, being the harmonic mean between $\mathscr{P}_k$ and $\mathscr{R}_k$, remains unchanged from the literature.

Intuitively, $\mathscr{P}_k$ relates to the proportion of energy *correctly* attributed to appliance $k$ relative to the *total* energy attributed by the algorithm to the same appliance. Whereas, $\mathscr{R}_k$ denotes the ratio between the correctly extracted energy and the actual energy consumed by appliance $k$. This means, a high wrongly extracted energy would result in a low energy-assignment precision, while a low energy-assignment recall is a consequence of failing to attribute energy to a given

appliance when it is actually being used. Ideally, both $\mathscr{P}_k$ and $\mathscr{R}_k$ for appliance $k$ should be high and well-balanced. Therefore, $\mathscr{F}_k$ can be used to gauge the overall correct extraction rate, taking into account both aspects of precision and recall.

### 5.5.2 Evaluation on Synthetic Data

We first consider the worst case in which known and unknown appliances have similar power consumption. To validate the significance of the duration-dependent component of RdFVTHMM in resolving the similarities in this regard, power consumption data of 3 synthetic appliances with the same distribution of power are generated for disaggregation purposes. Each appliance has two states – ON and OFF – and two of the appliances have the same appliance state duration distribution. The model parameters used for synthesising the data are summarised in Table 5.1 and Table 5.2. For each synthetic appliance, the generated data is aggregated together to form the aggregate data, which will be used as an input to the disaggregation process. A particular instance of the generated synthetic data is shown in Figure 5.12.

For disaggregation with RdFVTHMM and FVTHMM, the parameters used are exactly the same as those employed for generating the synthetic data, whereas for RdFHMM and FHMM, the Markov state transition matrices used have self-transition probabilities that are consistent with the mean state durations, i.e. $a_{i,i} = (E[d] - 1)/E[d]$. Table 5.3 shows the complete Markov state transition matrices derived in this manner.

In the experiment, one of the synthetic appliances will be set aside as the known appliance while the remaining two will be treated as appliances which

Table 5.1: Emission model of the synthetic appliances.

| **Synthetic Appliance** | **State,** $x_{t,k}$ | **Mean,** $\mu$ | **Standard Deviation,** $\sigma$ |
|---|---|---|---|
| 1, 2, 3 | 0 | 0.000 | 1.000 |
| | 1 | 50.000 | 4.000 |

Table 5.2: State duration model of the synthetic appliances.

| **Synthetic Appliance** | **State,** $x_{t,k}$ | **Shape,** $\alpha$ | **Scale,** $\beta$ |
|---|---|---|---|
| 1 | 0 | $4.500 \times 10^4$ | $6.667 \times 10^{-3}$ |
| | 1 | $1.333 \times 10^4$ | $1.500 \times 10^{-2}$ |
| 2, 3 | 0 | $2.500 \times 10^2$ | $2.000 \times 10^{-1}$ |
| | 1 | $1.440 \times 10^2$ | $1.667 \times 10^{-1}$ |

Table 5.3: State transition matrices used for disaggregation under RdFHMM and FHMM.

(a) Synthetic appliance 1

| **State,** $x_{t,1}$ | **0** | **1** |
|---|---|---|
| **0** | 0.996 | 0.004 |
| **1** | 0.005 | 0.995 |

(b) Synthetic appliance 2 and 3

| **State,** $x_{t,2}$ | **0** | **1** |
|---|---|---|
| **0** | 0.980 | 0.020 |
| **1** | 0.042 | 0.958 |

we have no model for. This means, the state inference algorithm will only have knowledge on the known appliance, even when the data to be disaggregated contains contributions from the other two unknown appliances. The role of the known appliance will be rotated between appliance 1, appliance 2 and appliance 3 to investigate the effect of the behaviour of the modelled appliance on the overall extraction process. The disaggregation accuracy is assessed using the precision and recall metric defined in Section 4.4.1, and comparison is made between RdFDHMM, RdFHMM, FDHMM and FHMM, in terms of their respective ability to deal with extreme cases of severe overlaps. For the robust models, a $\rho_\sigma$ of 1000 and a $\Delta_{\text{thres}}$ of 10 were used.

The extraction results for the synthetic data shown in Figure 5.12 are presented in Figure 5.13, with the corresponding disaggregation accuracies outlined in Table 5.4. From the figures, it can be seen that, when appliance 1 is the known appliance, RdFVTHMM-dPBDT performs the best with the highest precision and recall. This is because the state duration of appliance 1 is encoded in its model and the state inference algorithm is able to exploit clear differences in state duration characteristics to distinguish between the known appliance and unknown appliances. In contrast, without the duration component in the model, RdFHMM-dPBDT did not fare very well. In nearly half the time, it wrongly inferred the contribution of the unknown appliances as appliance 1 (see Figure 5.13b).

Also, as expected, both the non-robust methods (i.e. FVTHMM-PBDT and FHMM-PBDT) have poor results on average. It can be seen from Figure 5.14 that, whenever power is consumed, the known appliance is inferred to be the contrib-

Table 5.4: Comparison of different methods when applied to the generated synthetic data.

| **Precision (%)/Recall (%)** | | | |
|---|---|---|---|
| **Methods** | **Known Appliance** | | |
| | 1 | 2 | 3 |
| RdFVTHMM-dPBDT | 95.41/95.69 | 70.62/61.04 | 74.46/61.42 |
| RdFHMM-dPBDT | 31.57/22.40 | 48.05/79.93 | 66.39/69.24 |
| FVTHMM-PBDT | 56.06/96.28 | 42.67/95.55 | 45.02/96.00 |
| FHMM-PBDT | 56.06/96.28 | 42.67/95.55 | 45.02/96.00 |

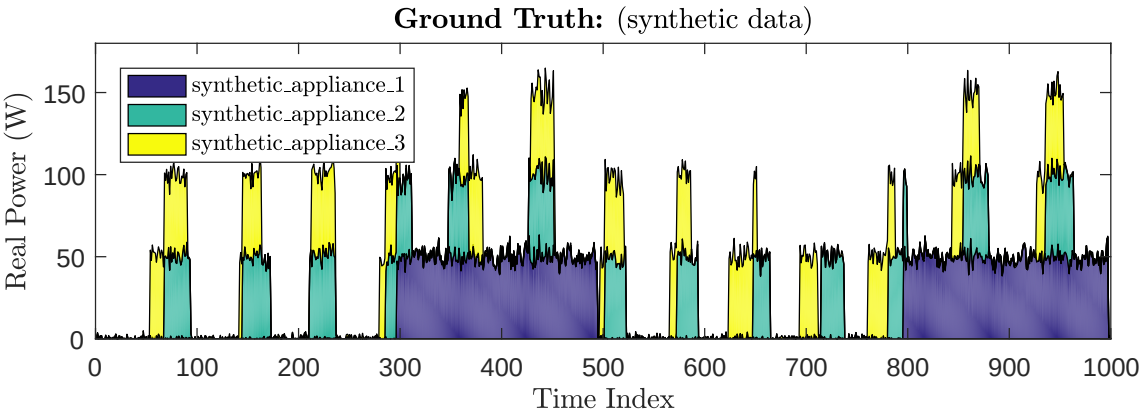
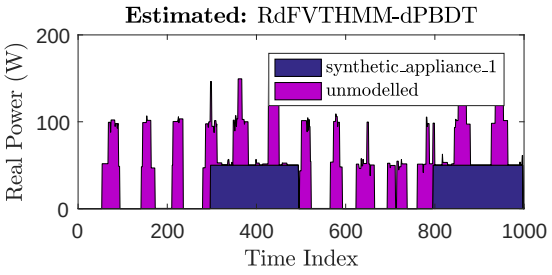
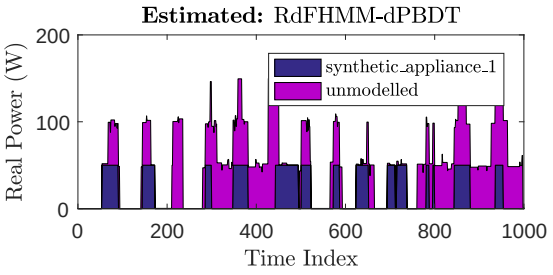


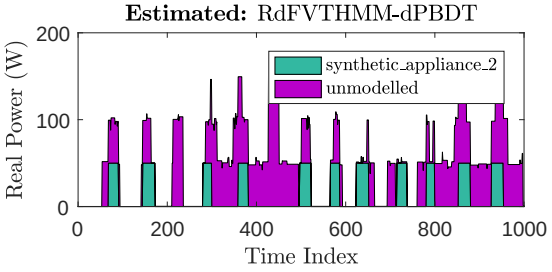

Figure 5.12: The generated synthetic data



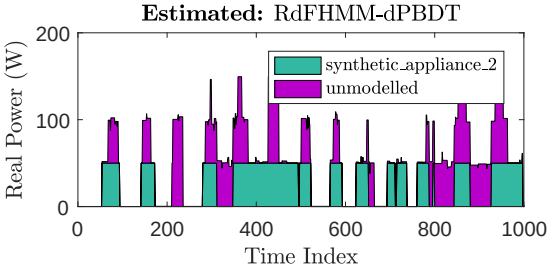

(a) Extraction of appliance 1 using RdFVTHMM-dPBDT.



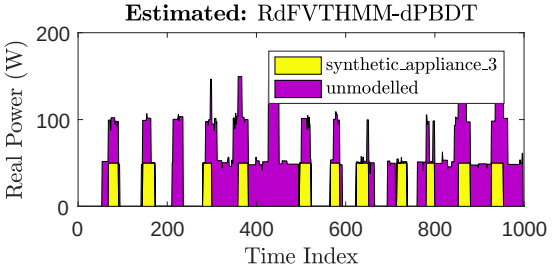

(b) Extraction of appliance 1 using RdFHMM-dPBDT.



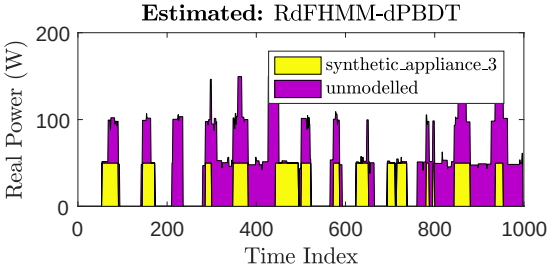

(c) Extraction of appliance 2 using RdFvTHMM-dPBDT.




(d) Extraction of appliance 2 using RdFHMM-dPBDT.




(e) Extraction of appliance 3 using RdFVTHMM-dPBDT.




(f) Extraction of appliance 3 using RdFHMM-dPBDT.

Figure 5.13: Comparison between RdFVTHMM-dPBDT and RdFHMM-dPBDT in extracting different known appliances.

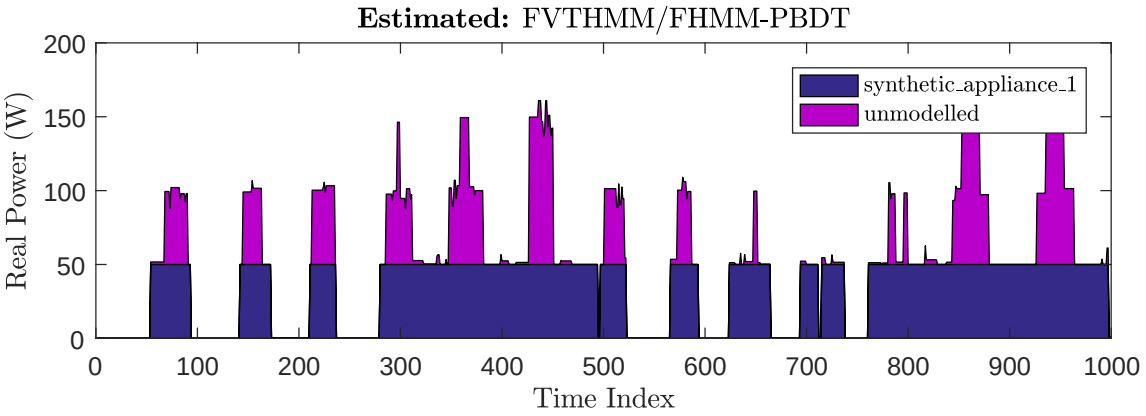



Figure 5.14: Extraction of appliance 1 using FVTHMM-PBDT or FHMM-PBDT.

utor, even though the observed value is in reality due to unknown appliances. This is because neither FVTHMM nor FHMM has the provision for unmodelled contributions to be ignored. Since appliance 1 is the only known appliance in the models used for state inference, the closest match (i.e. appliance 1) is simply chosen. The resulting assignment to appliance 1 in all time instances lead to many false positives and a low number of false negatives, producing the high recall and low precision shown in the last two rows of Table 5.4.

When either appliance 2 or 3 takes on the role of the known appliance, the RdFVTHMM-dPBDT method reduces in precision and recall. While this is not surprising, it highlights that whenever severe overlaps occur in all three aspects – state power distributions, Markov state transition probabilities and state duration distributions, correct identification becomes difficult. Unless more information is utilised (e.g. additional features), errors are to be expected. Fortunately, in a real-world setting, such cases are rare, with most instances being overlaps in the first two aspects, thus strengthening the usefulness of the state duration model in resolving similarities between known and unknown appliances.

### 5.5.3 Evaluation on Real-World Data

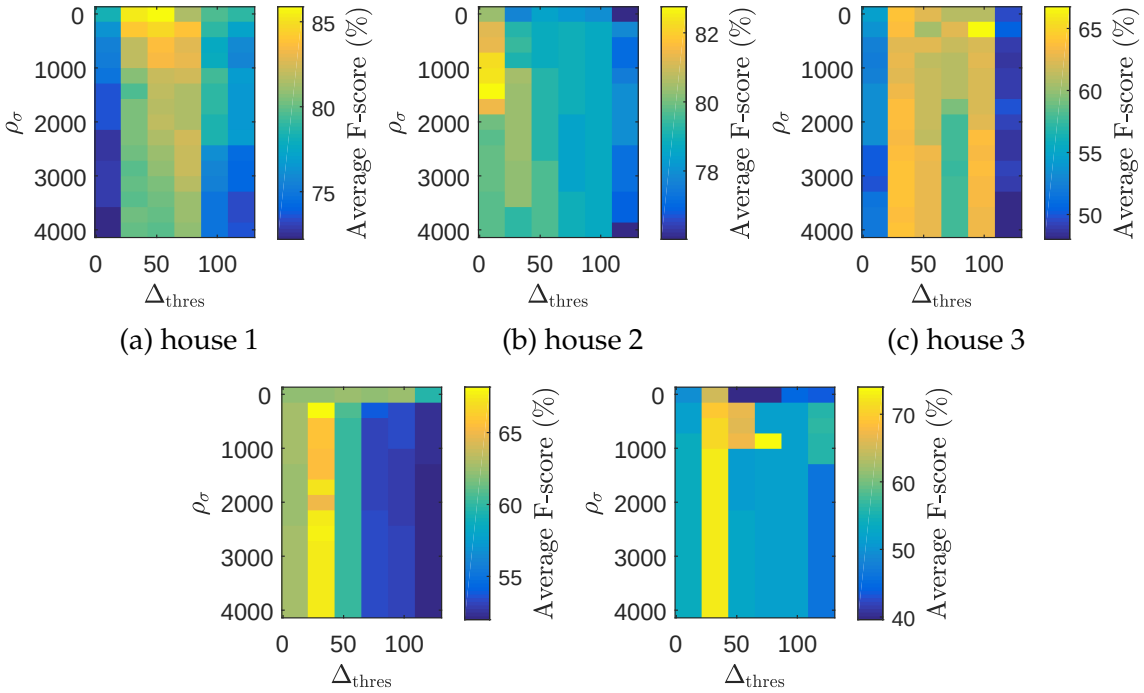
We have chosen to use the publicly available REDD dataset [KJ11] for evaluating how well the dPBDT algorithm with RdFVTHMM (RdFVTHMM-dPBDT) performs on real-world data. The time range of data used for testing is shown in Figure 4.18, while data outside the marked region is utilised in the training stage to learn appliance models. Similar to the evaluation done in Chapter 4, house 5 is not tested as plenty of data is missing (see Figure 4.17).

In the evaluation, three aspects of the proposed RdFVTHMM-dPBDT method are explored. Firstly, the influence of the flatness ratio $\rho_\sigma$ and the threshold $\Delta_{\text{thres}}$

on the extraction accuracy is investigated. Secondly, we evaluate and compare the method with other benchmark approaches. And lastly, we study how the number of modelled appliances to be jointly extracted could affect the extraction accuracy.

### The role of the flatness ratio $\rho_\sigma$ and the threshold $\Delta_{\text{thres}}$

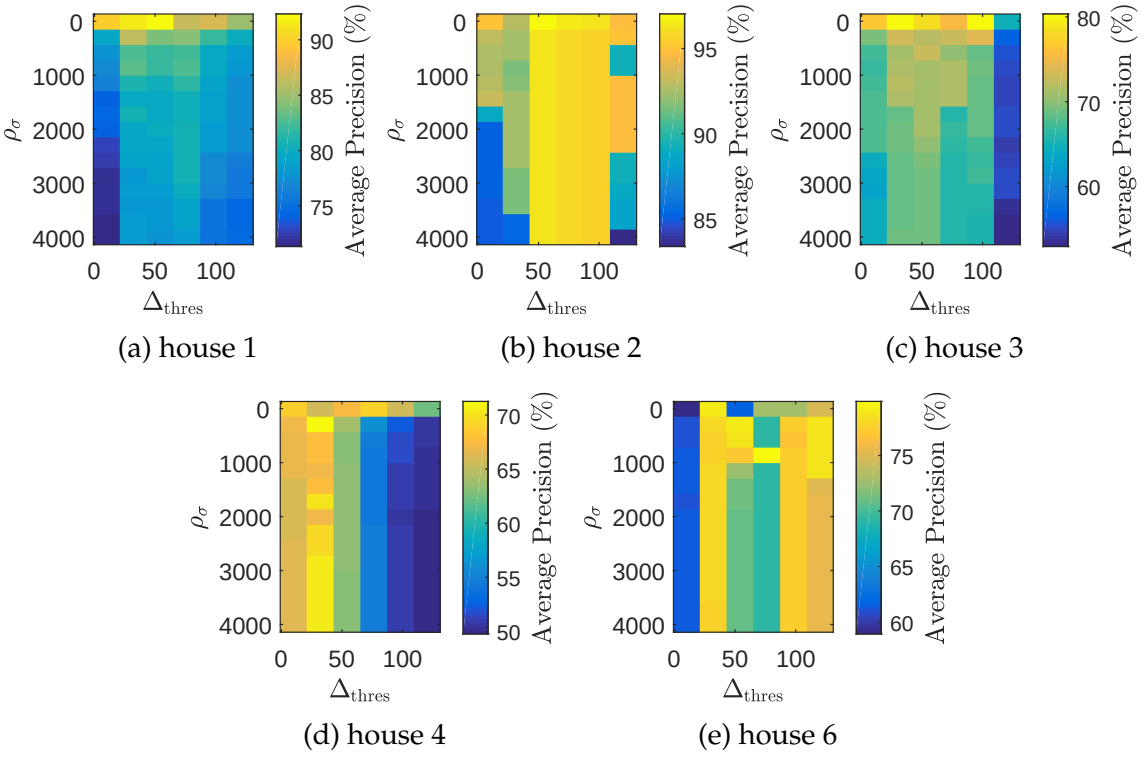
To examine the role of $\rho_\sigma$ and $\Delta_{\text{thres}}$ in impacting the extraction accuracy, RdFVTHMM-dPBDT is run multiple times on the test data of each house, each time with different values of $\rho_\sigma$ and $\Delta_{\text{thres}}$. We consider 15 evenly-spaced points of $\rho_\sigma$ from 10 to 4000 and 6 evenly-spaced points of $\Delta_{\text{thres}}$ from 10 to 120. In each round, one of the $(\Delta_{\text{thres}}, \rho_\sigma)$ pairs on this 6-by-15 lattice is used and the resulting average F-score is computed across all appliances to be extracted, i.e. $\overline{\mathscr{F}} = \frac{\sum_{k=1}^{K} \mathscr{F}_k}{K}$ with $K$ being the number of appliances in question. Also computed are the average precision, $\overline{\mathscr{P}}$, and average recall, $\overline{\mathscr{R}}$. The top 5 most energy-consuming appliances specific to each house from the training set are considered for extraction in each case (i.e. $K = 5$); the remaining appliances are unmodelled. $N_{\text{p,max}}$ is fixed at 100 throughout the whole experiment.




(a) house 1 (b) house 2 (c) house 3

(d) house 4 (e) house 6

Figure 5.15: Variation of the average F-score, $\overline{\mathscr{F}}$, for each house.



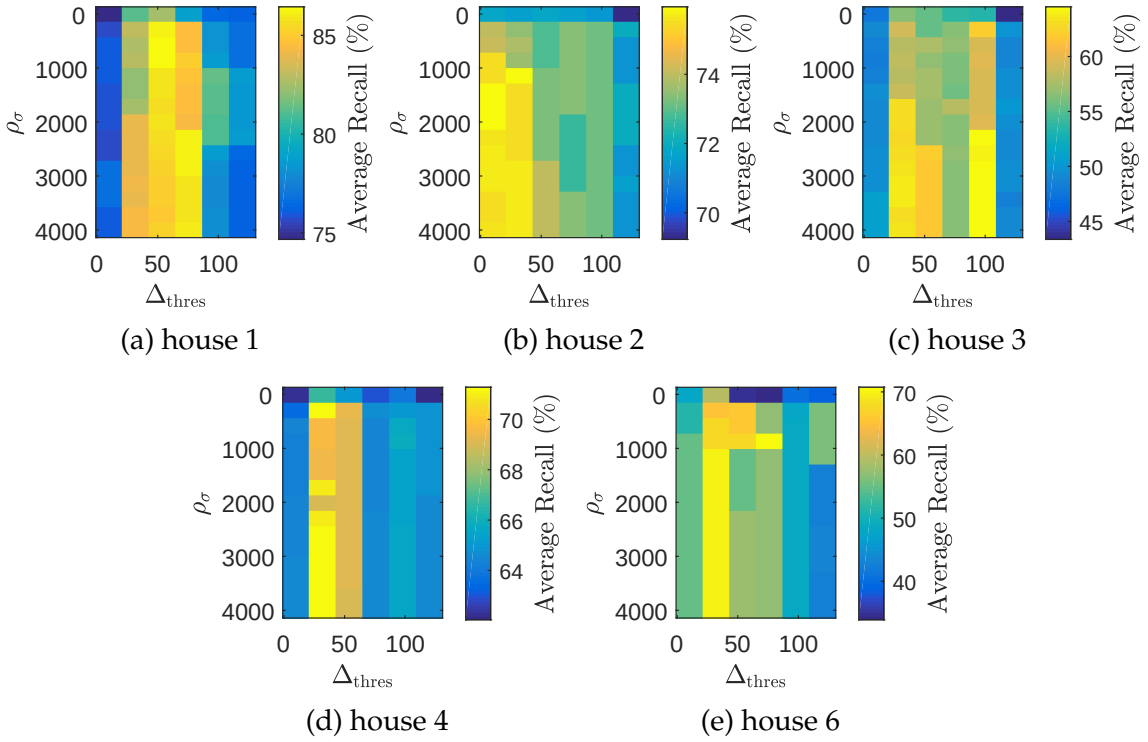

(a) house 1 (b) house 2 (c) house 3

(d) house 4 (e) house 6

Figure 5.16: Variation of the average precision, $\overline{\mathscr{P}}$, for each house.

Figure 5.15, Figure 5.16 and Figure 5.17 show the variation of the average F-score, the average precision and the average recall for each house over the lattice of $(\Delta_{\text{thres}}, \rho_\sigma)$, respectively. From the F-score figure, it can be seen that there is no single optimum pair, $(\Delta^*_{\text{thres}}, \rho^*_\sigma)$, which is common across all houses. Though not surprising given the different types of appliances contained in each house, the houses all seemingly agree that less extreme values for $\Delta_{\text{thres}}$ and $\rho_\sigma$ should be chosen. This is especially the case for $\Delta_{\text{thres}}$, since a value which is too large encourages important changes in steady-state power to be ignored, while a value which is too small allows non-stable power consumption values (due to transients) to be processed, thus affecting extraction accuracy negatively. In addition, as $\Delta_{\text{thres}}$ is also used as the maximum deviation from the moving average value before steady-state condition is violated, a small $\Delta_{\text{thres}}$ can prevent the steady-state segmentation algorithm from locking-in on stable measurements with large variance, incurring false negatives as a result.

As for the influence of $\rho_\sigma$, it was discovered that a large value generally results in a high recall but low precision (see Figure 5.17 and Figure 5.16), given that most unmodelled appliances in the evaluation are found to be in the low $<$ 200W range. In particular, when the emission probability term is dominant over the duration-dependent state transition term in the calculation of the parti-




(a) house 1 (b) house 2 (c) house 3

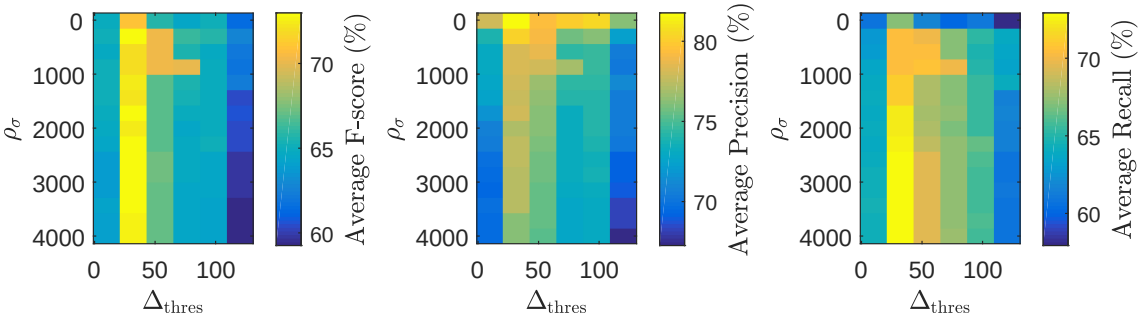
(d) house 4 (e) house 6

Figure 5.17: Variation of the average recall, $\overline{\mathscr{R}}$, for each house.

cle score (see (5.10)), the larger $\rho_\sigma$ is more likely to cause actual contributions of such low-powered unknown appliances to be assigned to modelled appliances mistakenly. Therefore, there is a higher false positive rate amongst the modelled appliances, leading to an overall lower precision. This is consistent with our description of $\Delta_{\text{thres}}$ and $\rho_\sigma$ in Section 5.3.2, where a large $\rho_\sigma$ implies a Laplace distribution which has its mass spread more widely. As such, the curves of the Gaussian distributions corresponding to modelled appliances tend to cover that of the Laplace distribution in the range of power consumption values closer to 0W (see Figure 5.6). Since most of the unmodelled appliances are low in power consumption, assignments of their contributions to modelled appliances are more likely to be favoured. Hence, a large $\rho_\sigma$ implies a higher recall at the expense of lower precision, as both Figure 5.17 and Figure 5.16 confirm.

By the same argument, because a small $\rho_\sigma$ results in a Laplace distribution with most of its mass concentrated around zero, modelled appliances with small power consumption are more likely to be attributed to unmodelled loads wrongly. However, the flip side is a lower chance of incurring a false positive. Accordingly, this means a generally higher precision in exchange for a lower recall.

The variation described thus far is more apparent if we take the mean of the average F-score, average precision and average recall over all houses, as shown in Figure 5.18. From this, it is clear that less extreme values of $\Delta_{\text{thres}}$ and $\rho_\sigma$ are preferred to reach a well-balanced trade-off between precision and recall.




(a) Mean of $\overline{\mathscr{F}}$ (b) Mean of $\overline{\mathscr{P}}$ (c) Mean of $\overline{\mathscr{R}}$

Figure 5.18: Variation of the mean of $\overline{\mathscr{F}}$, $\overline{\mathscr{P}}$ and $\overline{\mathscr{R}}$, computed across all houses.

### Comparison with baseline approaches

Similar to the evaluation done with the synthetic data, a few baseline approaches are compared against RdFVTHMM-dPBDT. We consider the original approach proposed in Chapter 4, FVTHMM-PBDT, and the variant without duration information, FHMM-PBDT. To gauge how the robust version of the former performs, RdFHMM-dPBDT is also included in the analysis. For both RdFVTHMM-dPBDT and RdFHMM-dPBDT, $\Delta_{\text{thres}}$ of 32 and $\rho_\sigma$ of 295 are used throughout this investigation by virtue of their maximum F-score shown in Figure 5.18a. As before, $N_{\text{p,max}}$ is fixed at 100 for all approaches, and only the top 5 most energy-consuming appliances from the training set are modelled and extracted; the remaining loads contributing to the aggregate measurements are unmodelled.

In terms of the correct assignment rate (CAR), the extraction accuracies of all approaches are summarised in Table 5.5, while Table 5.6 presents a more detailed outlook of how well the modelled appliances are extracted in terms of precision and recall. On average, the results indicate that RdFVTHMM-dPBDT outperforms the baseline methods. However, the outcome for house 6 is slightly surprising, as it is the only house where the non-robust methods have a higher CAR than that of the corresponding robust methods.

A closer look reveals a number of reasons. For **`outlets_unknown2`** in house 6, FVTHMM-PBDT and FHMM-PBDT have high recall but low precision (see Table 5.6). That is, false positives are prevalent but false negatives are un-

Table 5.5: CAR of different methods when applied to the REDD dataset

| CAR Metric (%) | | | | | | |
|---|---|---|---|---|---|---|
| **Methods** | **House** | | | | | **Average** |
| | 1 | 2 | 3 | 4 | 6 | |
| RdFVTHMM-dPBDT | 84.96 | 91.05 | 78.37 | 80.89 | 79.47 | 82.95 |
| RdFHMM-dPBDT | 75.51 | 86.91 | 76.07 | 79.09 | 75.20 | 78.56 |
| FVTHMM-PBDT | 57.87 | 83.72 | 64.92 | 67.52 | 85.83 | 71.97 |
| FHMM-PBDT | 56.63 | 72.24 | 64.13 | 43.50 | 77.12 | 62.72 |

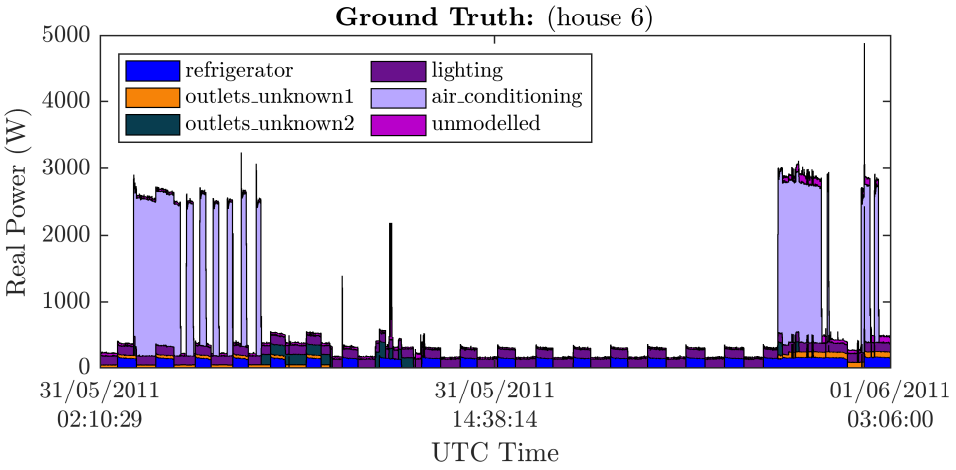
common. However, because CAR penalises wrong estimates of large power consumption devices more, by virtue of its formulation in (4.14), and because **air_conditioning**'s power consumption dwarfs that of **outlets_unknown2**, any wrong inferences pertaining to the former would mask the effect of wrong inferences related to the latter in the calculation of CAR. Seeing that **air_conditioning** is extracted with slightly better precision and recall by the non-robust methods while both robust methods have lower recall in comparison, the CAR is tipped towards the favour of FVTHMM-PBDT and FHMM-PBDT, regardless of whether or not the substantially lower power-consuming **outlets_unknown2** is extracted correctly.

As can be seen in Figure 5.19, there are in fact occasional misses by RdFVTHMM-dPBDT in the extraction of **air_conditioning**. Given that **air_conditioning** is the only device with power consumption beyond 2000W in house 6 and it should be the easiest to detect, this observation is especially surprising. Upon deeper investigation, it was found that the issue lies with the steady-state segmentation algorithm. In particular, the power consumption of **air_conditioning** during steady-state operation has a large variance relative to the employed $\Delta_{\text{thres}}$ of 32 (see Figure 5.20). Therefore, as alluded to previously in the previous investigation with regards to the use of small $\Delta_{\text{thres}}$, the steady-state segmentation algorithm is unable to lock-in on a stable value. In this regard, the contribution of **air_conditioning** during its steady-state interval is ignored and treated as originating from unmodelled loads mistakenly.

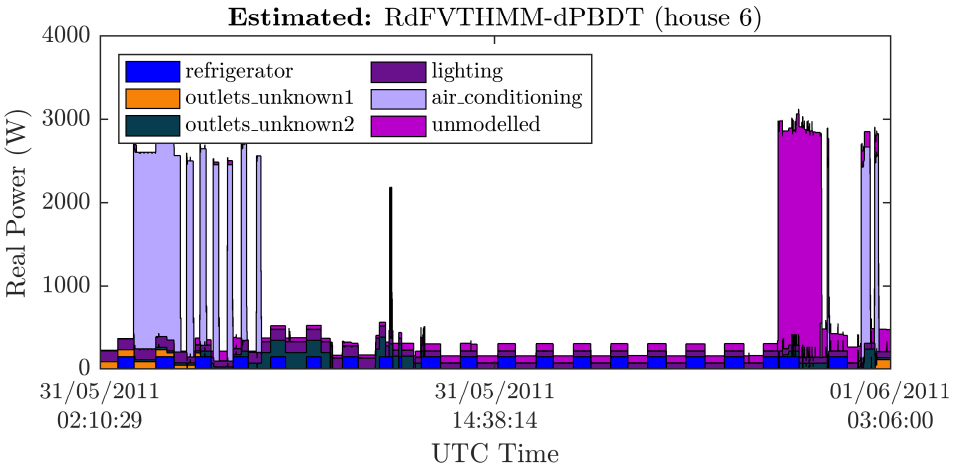
To confirm this hypothesis, RdFVTHMM-dPBDT is rerun on house 6 but now, with an increased $\Delta_{\text{thres}}$ of 76. The new CAR is 88.44%, a marked improvement over the previous case. In terms of the precision and recall of **air_conditioning**, both registered at 98.40% and 97.90%, respectively. However, the average precision and recall of house 6 dropped to 69.32% and 57.20%, owing to the lower precision and recall of **outlets_unknown1**: 28.78% and 0.33%. This is a result which is linked to the fact that the power consumption

Table 5.6: Precision and recall of different methods when applied to REDD dataset.

| House | Top 5 Appliances | Precision (%)/Recall (%) | | | |
|---|---|---|---|---|---|
| | | RdFVTHMM-dPBDT | RdFHMM-dPBDT | FVTHMM-PBDT | FHMM-PBDT |
| 1 | refrigerator | 97.21/91.23 | 89.03/90.58 | 67.94/71.62 | 65.94/75.26 |
| | dishwasher | 73.07/96.85 | 50.34/92.77 | 45.01/94.18 | 40.93/86.28 |
| | kitchen_outlets2 | 90.85/79.63 | 86.55/79.23 | 54.46/84.10 | 55.77/83.09 |
| | lighting1 | 76.14/71.69 | 62.07/61.09 | 44.44/70.11 | 43.52/64.36 |
| | washer_dryer3 | 94.23/75.88 | 76.05/98.69 | 72.67/99.48 | 72.67/99.48 |
| | **Average** | **86.30/83.05** | **72.81/84.47** | **56.90/83.90** | **55.77/81.69** |
| 2 | kitchen_outlets_1 | 81.15/22.07 | 29.04/24.55 | 77.60/79.31 | 46.41/83.64 |
| | lighting | 92.15/81.12 | 84.15/78.88 | 83.93/64.38 | 60.63/59.61 |
| | microwave | 93.36/73.67 | 88.32/65.90 | 93.51/90.13 | 64.51/88.97 |
| | kitchen_outlets_2 | 99.44/95.50 | 99.44/95.50 | 65.52/98.11 | 80.98/98.11 |
| | refrigerator | 96.05/94.86 | 92.27/94.03 | 82.64/95.02 | 79.34/76.00 |
| | **Average** | **92.43/73.45** | **78.65/71.77** | **80.64/85.39** | **66.38/81.27** |
| 3 | electronics | 78.86/87.04 | 82.56/88.03 | 51.35/91.72 | 51.12/92.69 |
| | refrigerator | 81.12/71.61 | 73.50/68.37 | 59.34/70.33 | 54.66/65.22 |
| | lighting2 | 60.76/33.59 | 44.04/32.18 | 40.18/54.72 | 36.06/53.40 |
| | washer_dryer | 84.53/77.02 | 83.89/77.02 | 71.37/77.40 | 86.24/76.18 |
| | lighting4 | 63.16/26.21 | 52.19/57.28 | 58.18/44.37 | 65.08/55.14 |
| | **Average** | **73.69/59.09** | **67.24/64.58** | **56.09/67.71** | **58.63/68.53** |
| 4 | lighting1 | 52.16/14.52 | 19.31/28.38 | 28.33/35.04 | 20.92/41.15 |
| | furnace | 84.17/85.25 | 86.72/78.89 | 80.18/70.77 | 75.04/52.14 |
| | stove | 57.75/91.04 | 63.51/83.87 | 35.93/93.06 | 20.13/92.55 |
| | lighting2 | 71.01/86.01 | 79.34/62.89 | 74.79/64.87 | 75.16/63.40 |
| | kitchen_outlets2 | 91.15/79.47 | 75.85/64.25 | 56.31/80.91 | 31.82/39.99 |
| | **Average** | **71.25/71.26** | **64.94/63.66** | **55.11/68.93** | **44.61/57.84** |
| 6 | refrigerator | 66.39/67.58 | 66.17/67.78 | 63.44/61.42 | 57.82/47.42 |
| | outlets_unknown1 | 77.40/94.94 | 22.58/3.27 | 74.62/66.06 | 72.96/85.76 |
| | outlets_unknown2 | 48.52/42.97 | 42.57/7.47 | 39.17/84.41 | 31.15/85.67 |
| | lighting | 98.97/57.52 | 98.87/58.51 | 97.61/90.08 | 96.23/68.14 |
| | air_conditioning | 98.46/64.05 | 45.64/45.76 | 97.97/97.94 | 98.40/97.94 |
| | **Average** | **77.95/65.41** | **55.17/36.56** | **74.56/79.98** | **71.31/76.98** |




(a) The ground truth for a day's worth of data from house 6 of the REDD dataset.




(b) Estimated using RdFVTHMM-dPBDT

Figure 5.19: False negatives associated with the extraction of **`air_conditioning`** from house 6.

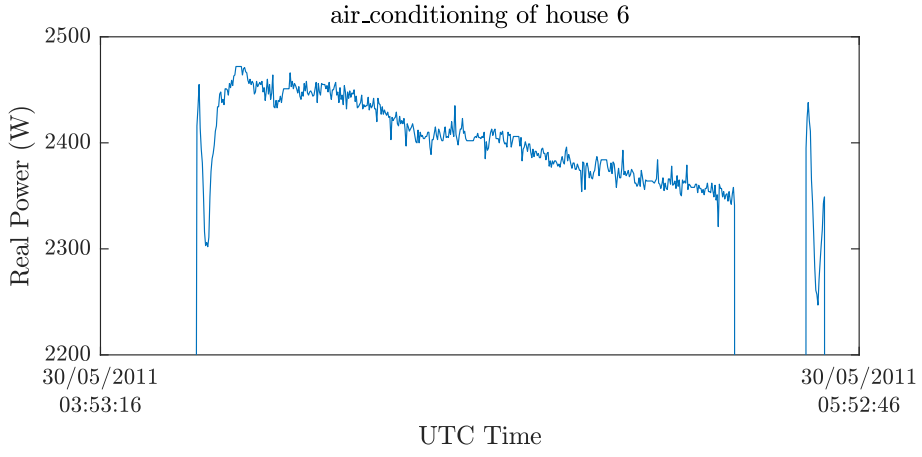



Figure 5.20: A closer look at the power consumption of **`air_conditioning`** from house 6.

of **outlets_unknown1** during operation is predominantly concentrated in the range from 40W to 120W, a portion of which falls below the $\Delta_{\text{thres}}$ of 76, and hence, the extremely low recall.

Further evidence on why this is due to the steady-state segmentation algorithm is the observation that both robust methods have low recall for **air_conditioning** whereas both non-robust methods have high recall and precision. As FVTHMM-PBDT and FHMM-PBDT do not include a steady-state segmentation procedure and both use the raw unprocessed aggregate power measurements as they arrived, the lower than expected recall of **air_conditioning** resulting from the weakness in the steady-state segmentation algorithm do not occur.

Overall, this suggests that a more sophisticated steady-state segmentation algorithm, which also dynamically takes into account changes in variances, is required. We envision its inclusion as part of RdFVTHMM-dPBDT, in place of the existing segmentation algorithm, would prevent issues like this from occurring, thus further improving extraction accuracy.

One other interesting observation that can be made with regards to **outlets_unknown2** is that it appears to correspond to many appliances instead of just one. This may explain why portions of its emission distribution which are significant in value, cover a large part of the power consumption domain between 0W and 500W (see Figure 5.21) and it may also be the reason why its distribution has multiple modes which are less well-defined. Therefore, modelling is a more challenging task and the fitted distribution may not be optimal for state inference, leading to the generally poor extraction results of **outlets_unknown2** across the methods considered.

It is worth noting that, with the exception of house 6, the precision and recall for the extraction of refrigerators by RdFVTHMM-dPBDT are consistently high across all houses (the **kitchen_outlets2** submeter of house 4 actually corresponds to a refrigerator, upon closer inspection). Being an appliance which is operating in a cyclic manner, this is not totally surprising, considering that the state duration model is able to capture this characteristic well. As for the refrigerator of house 6, there may be a number of factors contributing to the lower than expected precision and recall. The previously mentioned wide span of **outlets_unknown2**'s emission distribution could be a reason. However, more investigations need to be conducted to yield a definitive answer.

Also interesting from Table 5.6 is the observation that all non-robust methods have consistently lower average precision than average recall. This is a conse-

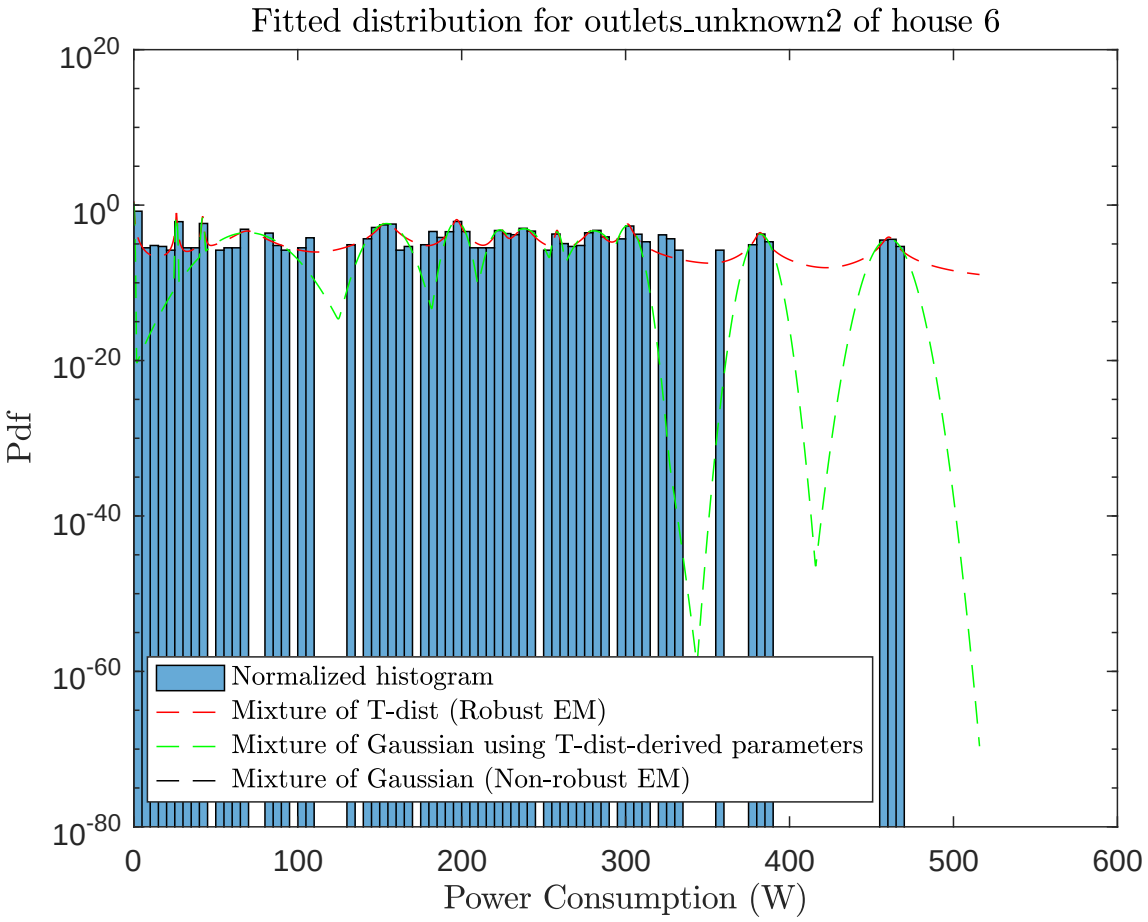



Figure 5.21: Emission model for **`outlets_unknown2`** of house 6 in the REDD dataset.

quence of absorbing actual contributions of unmodelled loads into estimated contributions of modelled appliances, in exchange for the unintentional side effect of lowering the likelihood of false negatives. Therefore, more emphasis should be placed on the precision metric than the recall metric, when modelled appliances are to be extracted in the presence of unmodelled loads.

### Number of appliances to extract

The number of modelled appliances to be jointly extracted, $K$, and its effect on the overall extraction accuracy were also studied. Like before, $N_{\text{p,max}}$ of 100, $\Delta_{\text{thres}}$ of 32 and $\rho_\sigma$ of 295 are employed for this investigation. For the experiment, $K$ is varied from 1 to 10 for each house, except for house 2, since it has a maximum of 7 appliances. Therefore, $K$ is only considered up to 7 for that particular case. At each round, the top $K$ most energy-consuming appliances are chosen to be modelled and extracted; the remaining loads are unmodelled and treated as appliances whose their characteristics are unknown.

Figure 5.22 shows variation in the average F-score, $\overline{\mathscr{F}}$, of RdFVTHMM as the number of appliances to be jointly extracted, $K$, increases. The overall trend is that the increase in $K$ results in a decrease in $\overline{\mathscr{F}}$. However, the downward trend for each house is not strictly monotonic. This may be explained by the decision that only the top $K$ most energy-consuming appliances are selected to be modelled and extracted, whereas there are many different combinations of $K$ appliances out of the maximum number of submeters in each house, $K_{\text{max}}$, in reality. As the selection of $K$ appliances in this way is only a single realisation out of the many different combinations, the inherent randomness might have contributed to the non-monotonicity. Another possible reason is, for some combination of appliances, having an additional modelled load may increase or reduce the potential confusion between competing solutions, depending on whether or not the newly modelled load is easier or harder to detect (see Figure 5.24). Nevertheless, the general downward trend is apparent if we consider the mean of the average

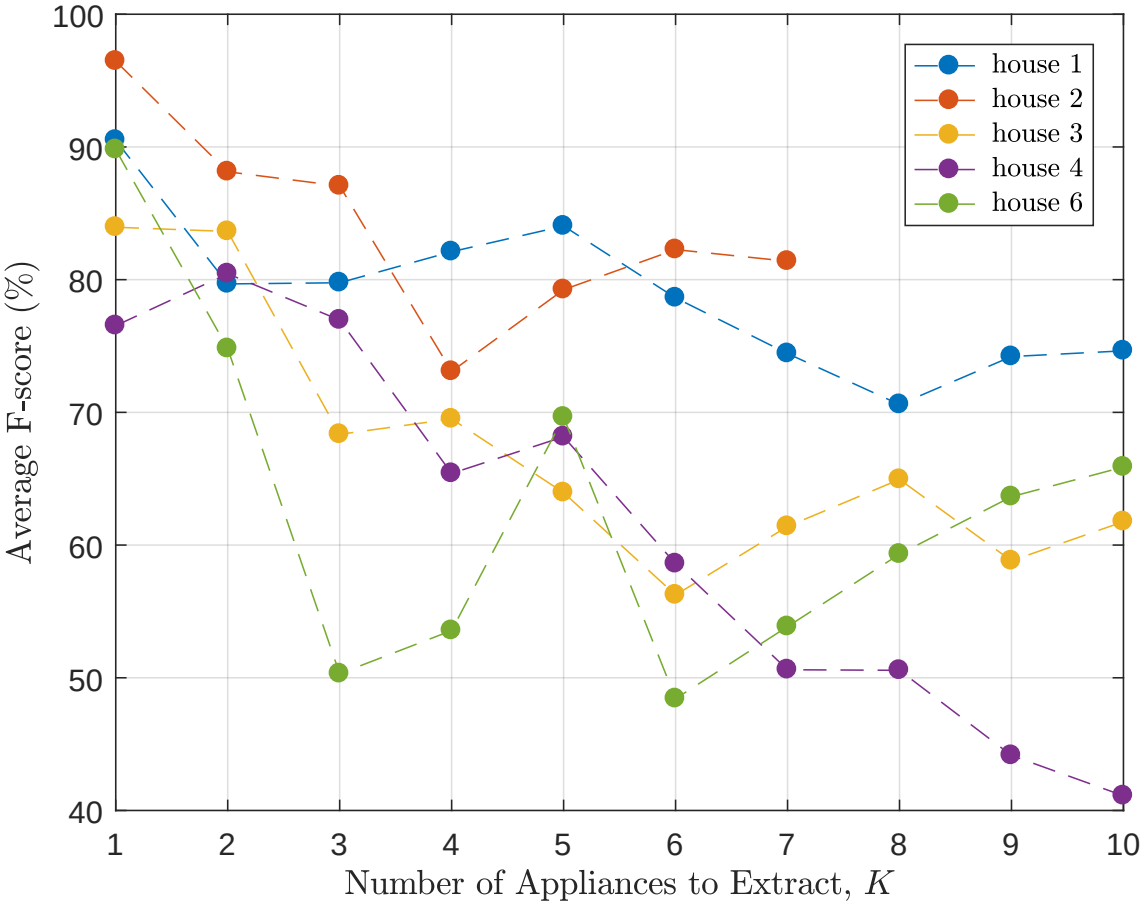



Figure 5.22: Average F-score, $\overline{\mathscr{F}}$, of RdFVTHMM-dPBDT against the number of appliances to extract, $K$. The top $K$ most energy-consuming appliances for each house is considered in each case.

F-score taken over all houses, as shown in Figure 5.23a. By itself, the decreasing trend could be attributed to the growth in the solution space as the number of appliances to extract increases; there is more potential for mix-ups in the state estimates of appliances which are modelled, leading to the likely case of more false positives and false negatives, as confirmed by the declining trend of both average precision and average recall in Figure 5.23b and Figure 5.23c respectively.

Interestingly, the corresponding average F-score for FVTHMM-PBDT shows an upward trend before appearing to taper off at the end, so does its average precision. If we recall that FVTHMM-PBDT does not explicitly account for the existence of unmodelled loads, this result is expected; contributions of power consumption, regardless of whether or not they are actually from unmodelled loads, get assigned by the algorithm to the closest matching modelled appliances. For small $K$, few modelled appliances absorbed the contributions of many unmodelled appliances, resulting in severe false positives. Early on, this dominates the effect of the growth in solution space, which is why an increase in average precision, instead of a decrease, is seen. As $K$ increases, the potential for false positives and false negatives due to assignments of unknown contributions to modelled appliances decreases, while that of the problem relating to more competing solutions begins to dominate. As a result, the rate of increase in average precision reduces before tapering off. It is hypothesised that after a certain $K$ beyond 10, the average precision of FVTHMM-PBDT would start to decrease. However, additional work is required to confirm this.

Relative to RdFVTHMM-dPBDT, the higher average F-score for FVTHMM-PBDT when $K \geq 7$ is largely attributed to the high recall in spite of the increase in the number of appliances to extract. While this may appear to show that RdFVTHMM-dPBDT does not perform as well as FVTHMM-PBDT, it is not true. In fact, the large recall is simply an artefact of the modelled appliances absorbing the power contributions unsparingly. Therefore, as mentioned at the end of the previous investigation, where comparisons with baseline approaches are made, more attention should be devoted to the precision metric when non-robust and robust counterparts are compared.

The appliance-wise F-scores of RdFVTHMM-PBDT for this investigation are also recorded, and they are presented in Figure 5.24. Several important observations can be made from the results. Firstly, it was found that the contributions of a number of appliances could be robustly estimated, even as the number of modelled appliances to be jointly extracted increases. In particular, refrigerators are generally able to maintain high F-scores. The same can be said for other devices

with large power consumption such as furnaces. This gives credence to the suggestion that future work on iterative disaggregation should first aim to subtract away inferred contributions of appliances which are known to be distinctive (e.g. refrigerators) and known a priori to draw large amount of power. Subtracting away these initial estimates from the aggregate measurements before extracting from the newly adjusted aggregate signal should provide better inferences of the remaining devices, especially small powered loads (e.g. lighting).

Secondly, the F-score associated with most appliances are relatively stable over the different values of $K$. This seemingly suggests that the proposed method, RdFVTHMM-dPBDT, is robust against the changing composition of the unmodelled loads. However, given that the result presented in Figure 5.24 is just one instance of choosing the $K$ appliances to be modelled and extracted, more investigations on how the results vary over all $\binom{K_{\max}}{K}$ combinations have to be performed as part of any future work.

## 5.6 Summary

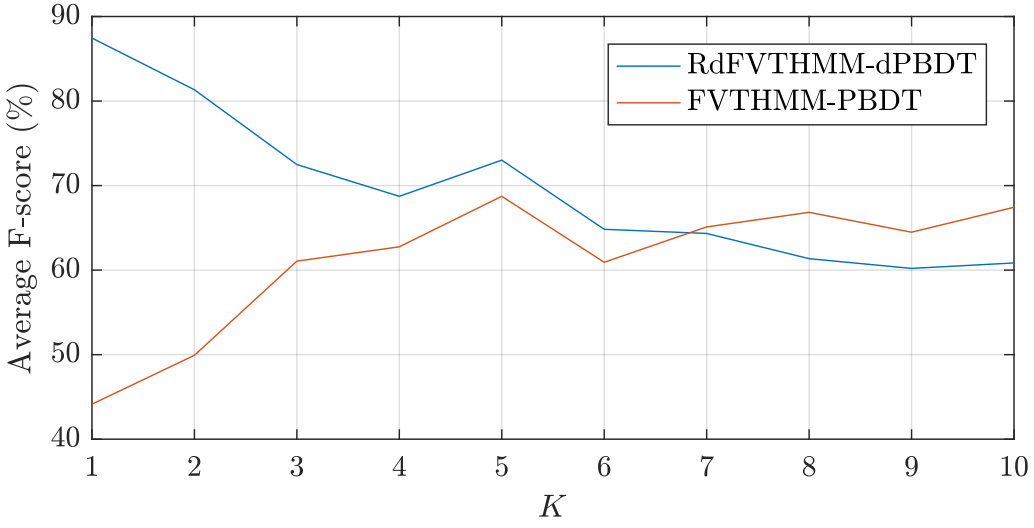
The techniques presented in this chapter have been shown to be beneficial for extracting the power contributions of appliances of interest, even in the presence of unknown loads. By combining the FVTHMM-PBDT framework described in Chapter 4, the noise model adapted from the field of compressed sensing, and a steady-state segmentation algorithm, the robust extension, RdFVTHMM-dPBDT, is able to infer the power consumption of modelled or known appliances accurately, while benefiting from the real-time and efficient computation afforded by the PBDT algorithm.

The method is also practically appealing, as there is no longer the requirement to obtain the models for each and every appliance in a residential unit before accurate disaggregation could be performed. Instead, all that is needed is to specify the models for a few important loads that should be detected and the approach is able to extract their power contributions from the aggregate measurements; the power contributions of the remaining unmodelled appliances are implicitly assigned to a robust mixture component whose change-in-values is imposed with a sparsity constraint.

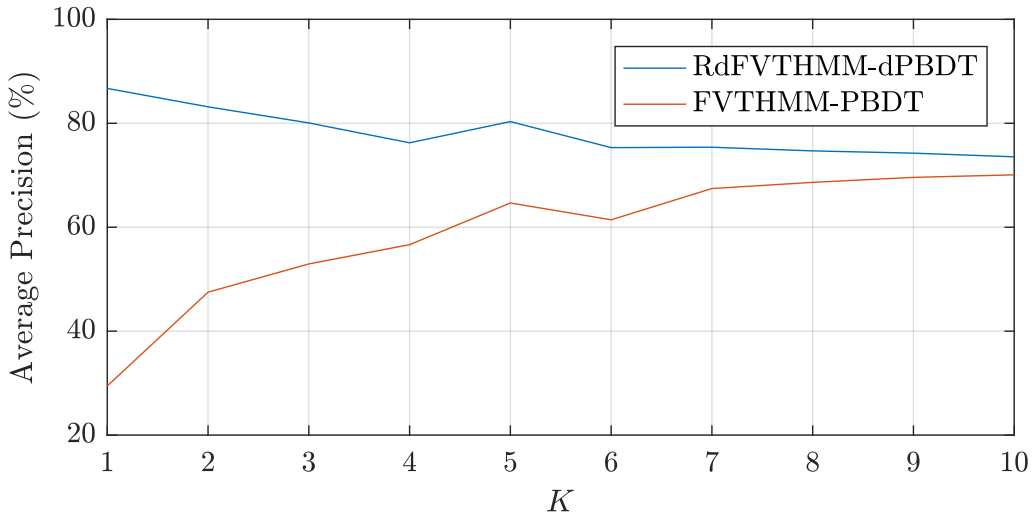
The evaluation of RdFVTHMM-dPBDT with synthetic data and real-world data has uncovered a number of significant results. Firstly, we have demonstrated that RdFVTHMM-dPBDT could distinguish between power values of modelled and unmodelled appliances which are similar, unlike the baseline approaches.

Secondly, robust extraction of the power contributions of modelled loads has been shown to be possible, as evident from the stable F-scores of many appliances when the composition of unmodelled loads changes.

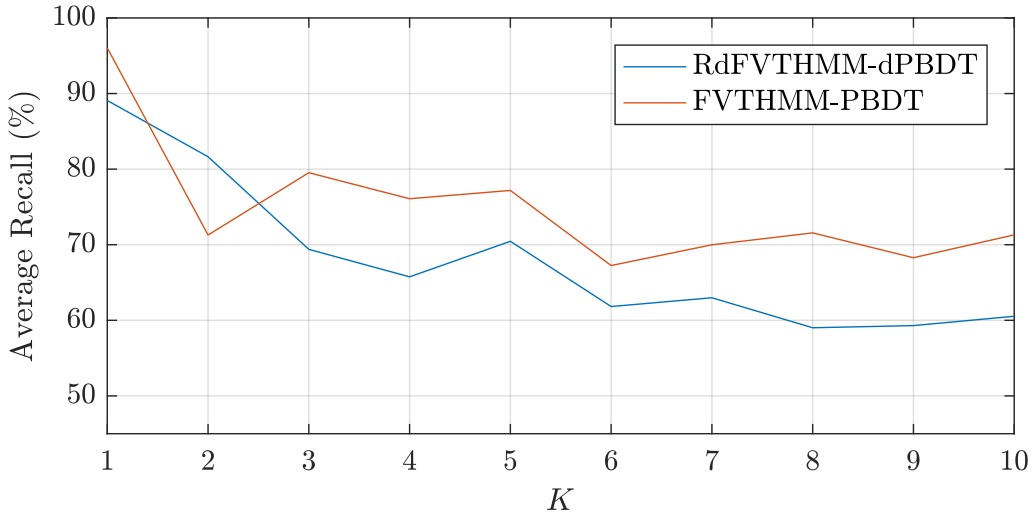
Although the disaggregation outcomes were in general excellent, it was found that errors could be further reduced if an improved procedure for the extraction of steady-state segments and the detection of state changes is used. Examples include an edge detector with a threshold that is adaptive, or perhaps, even a simple extension with a threshold that scales proportionally with the noise level and the magnitude of the aggregate measurements. However, such improvements are reserved for future work. Further, it is also hoped that more studies are made in relation to the systematic selection of the flatness ratio, and by extension, the rate parameter of the Laplace distribution governing the variation of the robust mixture component.




(a) Mean of $\overline{\mathscr{F}}$




(b) Mean of $\overline{\mathscr{P}}$




(c) Mean of $\overline{\mathscr{R}}$

Figure 5.23: Variation of the mean of $\overline{\mathscr{F}}$, $\overline{\mathscr{P}}$ and $\overline{\mathscr{R}}$, computed across all houses.

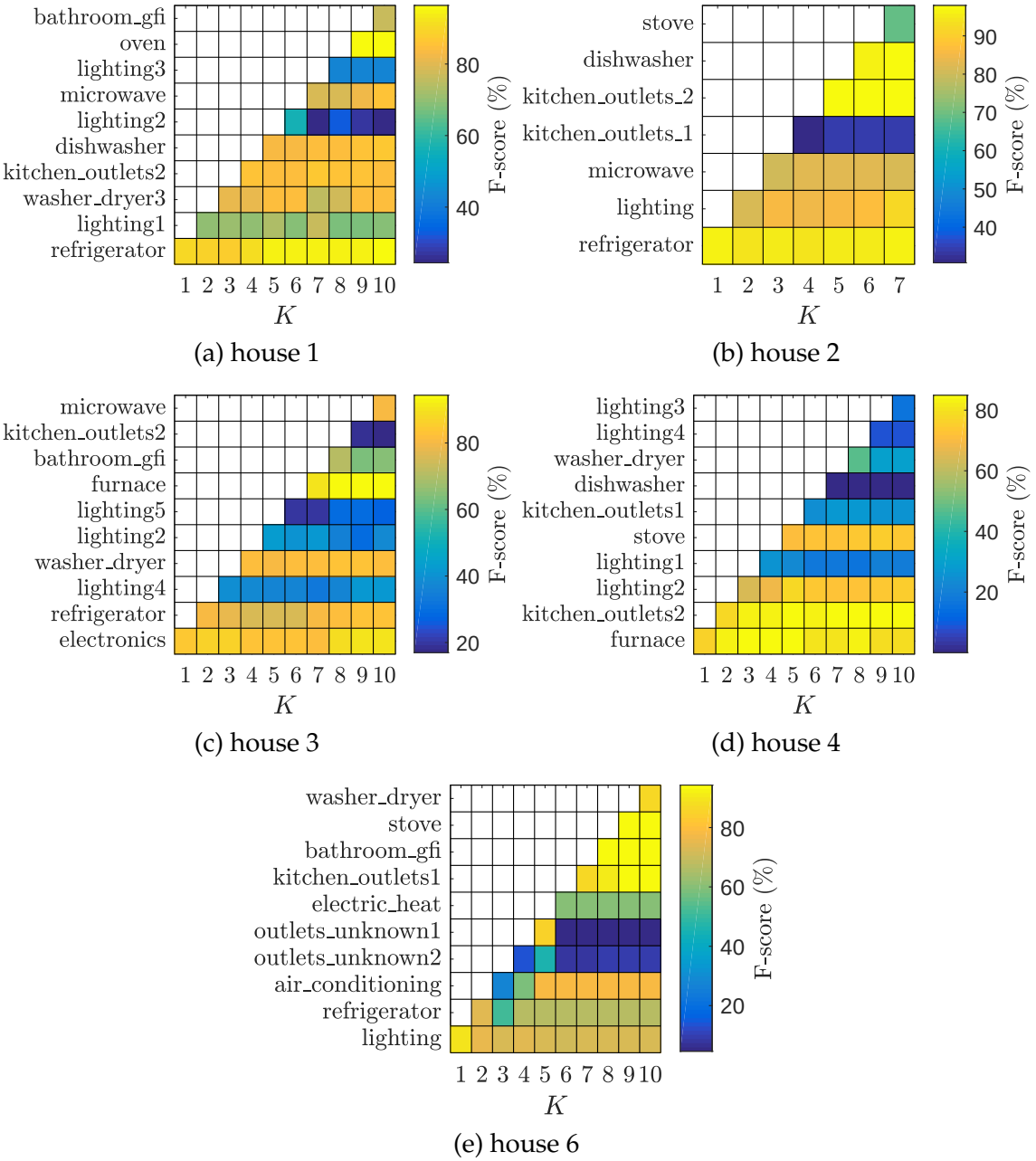



Figure 5.24: The appliance-wise F-score, $\mathscr{F}_k$, of RdFVTHMM-dPBDT as the number of appliances to be jointly extracted, $K$, increases.

CHAPTER SIX

# CONCLUSION AND FUTURE WORK

## 6.1 Conclusion

In this thesis, we have looked at the problem of Non-intrusive Load Monitoring (NILM) – a class of load-monitoring techniques by which whole-house/aggregate energy measurements are mathematically decomposed or disaggregated into per-appliance energy usage information. The basic premise is that, activations and deactivations of appliances can be detected as fluctuations in the aggregate signal, and the source of such signatures can be identified through a carefully-designed software algorithm.

In contrast to more hardware-centric methods whereby each appliance has its own a dedicated monitoring device attached, only a single sensor or energy monitor is required in NILM to determine the relative proportion of energy consumed by each appliance. As such, NILM is a non-invasive, cost-effective means of load monitoring and has been envisioned to be an important software counterpart to disaggregate whole-house energy consumption data from utility-installed smart meters, thereby making actionable information (e.g. itemised electricity bills) more widely accessible to homeowners, and empowering them with the ability to identify sources of energy wastage more accurately.

Although numerous NILM approaches have been proposed in the literature prior to the investigation conducted in this thesis, several important issues (as highlighted in Chapter 1 and Chapter 2) were still present to impede practical deployments and the widespread adoption of NILM in the real-world setting. For one, especially in situations where only *low*-frequency aggregate measurements are available (e.g. smart meters), there is a lack of study on how appliances with similar power signatures can be differentiated. Secondly, methods for *efficient* and *real-time* inference of appliance-level measurements under complex but more powerful models (needed to address the first issue) have been limited. Thirdly,

apart from the prominent work by Kolter and Jaakkola [KJ12], it is often assumed that all appliances in a residential unit can be modelled, and unknown or unmodelled loads (e.g. newly-installed appliances) are non-existent; the more challenging problem of extracting the power contributions of known appliances in the presence of outlying values owing to these unknown loads is rarely explored.

To that end, we have developed and presented a new robust real-time disaggregation framework in the preceding chapters of this thesis, with the following main contributions.

In Chapter 3, we have proposed an alternative variant of hidden semi-Markov model for representing appliance behaviour: factorial variable transition hidden Markov model (FVTHMM). It incorporates the duration information of states of appliances to resolve the aforementioned similarities in signatures. However, unlike existing hidden semi-Markov model used in NILM, such information is not used directly. Instead, the hazard function is employed as a time-varying, duration-dependent state transition probability, thus enabling the incremental calculation of probability values at any given time step. The significance of this is, it allows improved identification of appliances with similar power consumption, while providing a model formulation which is suitable for real-time disaggregation, as the experimental results have shown.

In Chapter 4, we presented a new tool – particle-based distribution truncation (PBDT) – that could be used to perform efficient computations and inferences on the more powerful model developed as part of Chapter 3. The method combines the survival-of-the-fittest concept of particle filters and the dynamic programming paradigm of the Viterbi algorithm, allowing estimates (represented by particles) at each time step to be tracked efficiently. By means of a number of heuristics and the sharing of computation results enabled by exploiting the distribution of particles, the PBDT algorithm is able to scale well computationally; experimental results illustrated that an average per-sample processing time of below 1 second is achievable for houses with as many as 20 billion states while attaining an average disaggregation accuracy of approximately 80%. In addition, we have demonstrated empirically that the time complexity of the algorithm is approximately linear in the number of appliances, validating its scalability and its usefulness for real-world NILM applications involving many appliances.

Following the disaggregation results of real-world data using PBDT for FVTHMM (i.e. FVTHMM-PBDT), Chapter 4 also investigated the benefits of explicitly modelling the non-stationary variation in the power measurements for a given state of an appliance. Specifically, we no longer assume that the mean

power consumption for a particular state is constant. Instead, it is now assumed to vary according to the state dwell time, much like the duration-dependent state transition probability. This allows gradual decreases in power consumption, like those seen during the ON cycles of refrigerators, to be used as features for improving disaggregation. In a preliminary study conducted, a disaggregation accuracy of 85% (an additional 5% from before) has been shown.

Finally, Chapter 5 looked at the problem of detecting modelled appliances in the aggregate measurements when unknown or unmodelled loads are present. We extended the base model, FVTHMM, proposed in Chapter 3 with a noise model that assumes sparse transitions for the unmodelled devices. Together with the addition of the change in power values as observed variables, the outcome is a robust version of FVTHMM, which we called RdFVTHMM. Further, due to a number of problems related to the direct application of the original PBDT algorithm under RdFVTHMM, a modified variant of the same algorithm, dPBDT, is formulated. In particular, it includes a steady-state segmentation procedure based on Hart's work [Har85] to reduce the sensitivity of the algorithm towards transient power values (e.g. power surge and slow rise-time). Additionally, the generation of particles is no longer performed at each time step. Rather, it is only done at times when states are inferred to change. All these allow the power contributions of appliances of interest (e.g. modelled appliances) to be robustly extracted from the aggregate measurements in real-time, without having to specify or learn the models of unknown loads. In the evaluation, experimental results validated the stability of the estimates for appliances whose behaviour is distinctive, confirming the robustness of the proposed disaggregation framework.

## 6.2 Future Research Directions

The aforementioned contributions have culminated in a robust disaggregation framework and have been shown to be promising in various aspects. The ability to extract power contributions of appliances efficiently in real-time while being resilient against perturbations owing to unknown loads is beneficial to the realisation of the many applications set out in Chapter 1. For example, new appliances, either introduced by guest visits or new purchases, can no longer severely affect the detection accuracy of existing appliances. Also noteworthy is the ability to perform tracking of appliance usage in real-time, opening the potential for various smart home use cases.

However, there are still open problems to be addressed and a number of improvements that could be made, opening avenues for future exploration. Therefore, in this section, equipped with the insights gained from this research, we propose several directions on which further work could be taken.

### Alternate Notion for the States of Appliances

For simplifying the modelling process, the work presented in this thesis and those of existing approaches in NILM [MHHE11, KJ11, EBE15, MPB+16, KDM+16] have employed the notion that a state of an appliance should correspond to a unique cluster of power values. However, this need not be the case in reality. As we have seen in Chapter 3, for example, the dishwasher consumes 0W for long periods of time when it is not being used, and it can also do the same for short time intervals, interleaved between non-zero power values within an operating cycle. This suggests that a distinction in states could be made for power values of 0W, depending on whether they are embedded within an operating cycle or not.

Likewise, the stove shown in Chapter 3 has a longer pulse of 400W at the start, followed by shorter pulses of the same magnitude, whenever it is being operated. Considering that these observations physically correspond to the process of the initial heating phase and the subsequent phase where the temperature is being regulated, it seems appropriate to introduce another state for differentiating between the two 400W observed at different parts of the operating cycle.

While we speculate that doing so may improve the detection of appliances working according to the finite state machine principle, introducing more than one state per power level in such a way would inflate the state space of the model significantly, potentially slowing down the speed of the state inference process. We believe, however, that the PBDT algorithm developed as part of this thesis would facilitate this alternate notion of states to be realised, given its efficiency and scalability when performing state inference over a large state space.

### Improved Emission Model

In Chapter 4, it was noted that the observed disaggregation errors are largely a consequence of the difficulty in modelling the variation in power consumption accurately. In particular, some appliances have per-state power consumption distribution that does not reflect the Gaussian assumption well; the underlying distribution is skewed with possible heavy tails. While we have partially addressed this by proposing the use of a Gaussian distribution whose mean varies exponen-

tially with the state dwell time, there is much scope for more work to be done in exploring other forms apart from exponential, the inclusion of variances that are also dependent on the state dwell time, and the integration with RdFVTHMM proposed in Chapter 5. Additionally, it is interesting to consider non-parametric variants based on Gaussian Processes [Ras04] for learning the appropriate forms from the training data.

Such a development is not only important for reducing errors but also could be useful in the tracking of continuously-variable loads, of which examples include heating, ventilation and air-conditioning (HVAC) systems and power drills. Even though these appliances are relatively uncommon in residential settings, it is envisioned that a generalised segmental emission model formulated in such a way, could spur future research on NILM systems targeted towards commercial and industrial applications. Further, the outcome might be an attractive alternative to the work of Laughman et al. [LKC+03], where current waveform data collected at high sampling rates is required to calculate the waveform harmonics necessary for extracting continuously-variable loads.

In more extreme cases, we can also use a hybrid generative-discriminative version of FVTHMM, where the appliance state transitions are still governed by the duration-dependent state transition probabilities in the generative sense but the emission model is now of the discriminative form. Doing so removes the burden of having to specify how the power consumption is distributed, reducing disaggregation errors due to deviations from modelling assumptions. A great example of how this could be achieved is the use of multilayer perceptron for directly learning a function whose input is the aggregate power consumption of a certain time step and the output is the state estimate of each appliance involved.

One other possibility is to use the Cauchy distribution for the emission model instead of the Gaussian distribution. This should solve issues related to heavy-tailed observations.

#### Fully Unsupervised or Semi-supervised Learning of Model Parameters

In Chapter 3, the learning of model parameters for FVTHMM is conducted with the assumption that training data in the form of appliance-level power measurements are available. In the real-world, however, the availability of such data may be limited and model parameters have to be inferred from only the aggregate measurements (i.e. completely unsupervised). For this reason, future efforts could be devoted to the formulation of a complete Expectation-Maximisation (EM) algorithm for FVTHMM, and a Monte Carlo implementation of the learning

procedure, given the intrinsic computational intractability of the said EM algorithm under FVTHMM.

Alternatively, the work by Johnson and Willsky [JW13] for the factorial version of the explicit duration HMM (FEDHMM) could be adapted and used for learning the model parameters in an unsupervised manner, since EDHMM can be transformed into an equivalent VTHMM [Joh05]. Therefore, the FVTHMM can be represented as a FEDHMM and unsupervised learning could be performed using the existing approach provided by Johnson and Willsky [JW13], while during disaggregation, online estimation of states could be done under FVTHMM, combining the best of both approaches.

Yet another way forward is to use a semi-supervised approach like in the work by Parson et al. [PGWR14], together with our proposed FVTHMM. Generic model parameters for classes of appliances whose behaviours can be generalised are learned over a diverse set of training data with similar devices of different brands. This allows a single common set of model parameters to be used across many houses, with no house-specific training data needed during the actual deployment of a NILM system; the generic model parameters are automatically tuned to house-specific model parameters using portions of the aggregate measurements where the individual power consumptions of a given appliance can be reliably extracted (e.g. during the night when less appliances are actively used). However, this can be further improved upon by using the proposed robust framework discussed in Chapter 5, potentially allowing power contributions of appliances to be extracted reliably at most times, even when the activity rate of appliances is high during the day. As such, the combination of the work of Parson et al. [PGWR14] and ours is an interesting task to pursue in the future.

**Iterative Disaggregation using the Developed Robust Extraction Scheme**

Iterative disaggregation is a concept that has been mentioned by Wong et al. [WWDc13] and Parson et al. [PGWR12]. The power measurements of appliances are not jointly extracted from the aggregate data. Instead, they are extracted iteratively by successive subtractions of the aggregate measurements. For example, inferred power contributions of an appliance is subtracted from the aggregate measurements and at each round, the aggregate measurements become simpler, facilitating further extractions of the remaining appliances. While the concept is interesting and should allow better detection of low-powered devices which are overwhelmed by the noise levels of other high-powered loads, the development of iterative approaches has been limited. We believe that the robust method pre-

sented in Chapter 5 could be used as the basis for future implementations of a robust iterative disaggregation technique, whereby the power contributions of appliances are extracted and subtracted in the same way that the idea was originally defined in [WWDc13] and [PGWR12] but in a substantially more robust and systematic manner.

**Methods for Fusion Point Tracking in PBDT**

In Chapter 4, the PBDT algorithm has been introduced to be a real-time and computationally efficient method for state inferences under powerful and complex models such as the proposed FVTHMM. Although we have provided the theoretical foundations in which backtracking can be performed from points in time where ancestors of descendant particles are common (i.e. fusion points), more work could be done for exploring efficient methods for tracking fusion points. Among others, a few important questions are

- How often do we search for fusion points?
- Should the method search for fusion points every time a new set of particles is generated?
- Can a rule be learned dynamically from the variation in the historical time lag data between the fusion point and the current time step?

Interestingly, these questions are closely related to the problem of garbage collection in computer science [WJNB95]. An example is the frequency in which memory spaces that are no longer referenced by a program are automatically identified and freed.

**Incremental Hashing in PBDT**

We have used the MurmurHash3 algorithm [App16] for hashing the extended system states and the extended device state in our implementation of PBDT. As multiple hashing operations are performed in each time step and the elements of the counter vector contained in a particle are mostly an increment of those contained in the parent particle, it may be beneficial to consider the use of incremental hashing algorithms [BGG94, BM97]. The rationale is that, previously computed hash values (e.g. for the parent particle) could be used to incrementally calculate the hash values for the current particle more efficiently, with potential for further improved computational performance of the PBDT algorithm.

### A Parallel PBDT Algorithm

For practitioners, it may be of interest to implement PBDT as a parallel algorithm, given that a group of parent particles is independent with one another (see Chapter 4), as far as the generation of new particles is concerned. Therefore, simultaneous and independent processing, with a multi-threaded implementation or a hardware chip-level implementation using field-programmable gate arrays (FPGAs), is worthwhile for further reductions in runtime and further improvements in computational scalability.

APPENDIX ONE

# DERIVATIONS FOR MML

This appendix provides the derivations of the message length expression and the formulation of the Expectation-Maximisation (EM) algorithm for minimising the message length as used in Chapter 3. For the problem considered, $S$ duration data points, $[d_s]_{s=1}^{S}$, of state $i$ of a certain appliance are modelled with a mixture of $L_i$ Gamma distributions. The value of $L_i$ and the corresponding parameters $\Theta_i$ governing the mixture model are unknown. They are to be estimated using the minimum message length (MML) principle; values of $L_i$ and $\Theta_i$ are chosen such that the length of the message consisting of the model parameters $\Theta_i$ and the encoded data $[d_s]_{s=1}^{S}$ is minimised.

## A.1 Message Length Formulation

In MML, the message composed of two parts, namely, the model parameters $\Theta_i$ and the data $[d_s]_{s=1}^{S}$ encoded using $\Theta_i$. As such, the generic expression of the total message length is

$$I(\Theta_i, [d_s]_{s=1}^{S}) = I(\Theta_i) + I([d_s]_{s=1}^{S} \mid \Theta_i), \tag{A.1.1}$$

where $I(\Theta_i)$ and $I([d_s]_{s=1}^{S} \mid \Theta_i)$ are the lengths of the respective parts.

The message length is simply the information content of the message, with additional terms resulting from the precision used for encoding the model parameters and the data points (see [WF87] and [KA15]). Accordingly, the message lengths for the encoded data and the model parameters are

$$I(\Theta_i) = \frac{p}{2} \log q_p - \log \left( \frac{p(\Theta_i)}{\sqrt{\left| \mathcal{F}(\Theta_i) \right|}} \right) \tag{A.1.2}$$

$$I([d_s]_{s=1}^{S} \mid \Theta_i) = -S \log \epsilon + \frac{p}{2} - \sum_{s=1}^{S} \log(p(d_s \mid \Theta_i)), \tag{A.1.3}$$

where $p$ is the number of free parameters in the model, $q_p$ is the lattice quantisation constant in $p$-dimensional space (see [CS84]), $\epsilon$ is the precision of the encoded data points and $\left|\mathcal{F}(\Theta_i)\right|$ is determinant of the Fisher information matrix. Note that we have chosen to use the natural log instead of log of base 2. Therefore, the message lengths have units of nats as opposed to bits.

To simplify the subsequent derivations, terms that are independent of $\Theta_i$ are denoted by constants, $\mathrm{C}_1$ and $\mathrm{C}_2$, such that

$$I(\Theta_i) = -\log\left(\frac{p(\Theta_i)}{\sqrt{\left|\mathcal{F}(\Theta_i)\right|}}\right) + \mathrm{C}_1 \tag{A.1.4}$$

$$I([d_s]_{s=1}^{S} \mid \Theta_i) = -\sum_{s=1}^{S} \log(p(d_s \mid \Theta_i)) + \mathrm{C}_2. \tag{A.1.5}$$

For $I([d_s]_{s=1}^{S} \mid \Theta_i)$, each of the summands in the first term of the right-hand side in (A.1.5) is the log likelihood of $\Theta_i$ given an observed data point $d_s$; the likelihood is characterised by a mixture of $L_i$ Gamma probability density functions, $g(d; \alpha_{l,i}, \beta_{l,i})$, i.e.

$$\begin{aligned} p(d_s \mid \Theta_i) &= \sum_{l=1}^{L_i} m_{l,i} g(d; \alpha_{l,i}, \beta_{l,i}) \\ &= \sum_{l=1}^{L_i} m_{l,i} \frac{d_s^{\alpha_{l,i}} \exp(-d_s/\beta_{l,i})}{\beta_{l,i}^{\alpha_{l,i}} \Gamma(\alpha_{l,i})}, \end{aligned} \tag{A.1.6}$$

where $\alpha_{l,i}$ is the shape parameter, $\beta_{l,i}$ is the scale parameter and $m_{l,i}$ is the mixing coefficient of the $l$th mixture component. Hence, the message length for the encoded data points resulting from the use of a given $\Theta_i$ is

$$I([d_s]_{s=1}^{S} \mid \Theta_i) = -\sum_{s=1}^{S} \log\left(\sum_{l=1}^{L_i} m_{l,i} g(d; \alpha_{l,i}, \beta_{l,i})\right) + \mathrm{C}_2. \tag{A.1.7}$$

On the other hand, to derive $I(\Theta_i)$, we follow the work of Oliver et al. [OBW96] and the work of Kasarapu and Allison [KA15] in approximating $\left|\mathcal{F}(\Theta_i)\right|$ as the product of the determinant of the Fisher information matrix for the each

mixture component and each model parameter. If $\Theta_i = [m_{l,i}, \alpha_{l,i}, \beta_{l,i}]_{l=1}^{L_i}$, then

$$\left|\mathcal{F}(\Theta_i)\right| \approx \prod_{l=1}^{L_i} \left|\mathcal{F}(m_{l,i})\right| \left|\mathcal{F}(\alpha_{l,i})\right| \left|\mathcal{F}(\beta_{l,i})\right|. \tag{A.1.8}$$

Like in [KA15], we also assume that the model parameters are mutually independent. Therefore,

$$\begin{aligned} I(\Theta_i) &= I([m_{l,i}, \alpha_{l,i}, \beta_{l,i}]_{l=1}^{L_i}) \\ &= \underbrace{\sum_{l=1}^{L_i} \log\left(\frac{p(m_{l,i})}{\sqrt{\mathcal{F}(m_{l,i})}}\right)}_{I([m_{l,i}]_{l=1}^{L_i})} + \underbrace{\sum_{l=1}^{L_i} \log\left(\frac{p(\alpha_{l,i})}{\sqrt{\mathcal{F}(\alpha_{l,i})}}\right)}_{I([\alpha_{l,i}]_{l=1}^{L_i})} \\ &\quad + \underbrace{\sum_{l=1}^{L_i} \log\left(\frac{p(\beta_{l,i})}{\sqrt{\mathcal{F}(\beta_{l,i})}}\right)}_{I([\beta_{l,i}]_{l=1}^{L_i})} + \mathrm{C}_1. \end{aligned} \tag{A.1.9}$$

According to [WB68] and [KA15], $I([m_{l,i}]_{l=1}^{L_i})$ could be further simplified to

$$I([m_{l,i}]_{l=1}^{L_i}) = \frac{L_i - 1}{2}\log(S) - \frac{1}{2}\sum_{l=1}^{L_i}\log(m_{l,i}) - \log(L_i - 1)! \tag{A.1.10}$$

For $I([\alpha_{l,i}]_{l=1}^{L_i})$ and $I([\beta_{l,i}]_{l=1}^{L_i})$, we use the results from the work of Agusta and Dowe [AD03],

$$\left|\mathcal{F}(\alpha_{l,i}, \beta_{l,i})\right| = \frac{S^2}{\beta_{l,i}^2}\left(\alpha_{l,i}\psi^{(1)}(\alpha_{l,i}) - 1\right), \tag{A.1.11}$$

where $\psi^{(u)}(\alpha_{l,i})$ is the $u$th order polygamma function defined as

$$\psi^{(u)}(\alpha_{l,i}) = \frac{d^{u+1}}{d\alpha_{l,i}^{u+1}}\log\left(\Gamma(\alpha_{l,i})\right). \tag{A.1.12}$$

Also, from the same work, the prior probability for $\alpha_{l,i}$ is assumed to be

$$p(\alpha_{l,i}) = \frac{2}{\pi(1 + \alpha_{l,i}^2)} \tag{A.1.13}$$

over the support of $(0, \infty]$, whereas the prior probability for $\beta_{l,i}$ is taken to be

$$p(\beta_{l,i}) = \frac{1}{\beta_{l,i}} \tag{A.1.14}$$

over the support of $[\exp(-8), \exp(8)]$. Altogether, after substituting the terms, we get

$$\begin{aligned} I(\Theta_i) = & \frac{L_i - 1}{2} \log(S) - \frac{1}{2} \sum_{l=1}^{L_i} \log(m_{l,i}) - \log(L_i - 1)! \\ & - \sum_{l=1}^{L_i} \log\left(\frac{1}{\beta_{l,i}}\right) - \sum_{l=1}^{L_i} \log\left(\frac{2}{\pi(1+\alpha_{l,i}^2)}\right) \\ & + \frac{1}{2} \sum_{l=1}^{L_i} \log\left(\frac{S^2}{\beta_{l,i}^2}\left[\alpha_{l,i}\psi^{(1)}(\alpha_{l,i}) - 1\right]\right) + \mathrm{C}_1, \end{aligned} \tag{A.1.15}$$

like shown in (3.31) of Chapter 3.

With both $I([d_s]_{s=1}^S \mid \Theta_i)$ and $I(\Theta_i)$ specified in A.1.7 and A.1.15 respectively, the total message length $I(\Theta_i, [d_s]_{s=1}^S)$ is

$$\begin{aligned} I(\Theta_i, [d_s]_{s=1}^S) = & \frac{L_i - 1}{2} \log(S) - \frac{1}{2} \sum_{l=1}^{L_i} \log(m_{l,i}) - \log(L_i - 1)! \\ & - \sum_{l=1}^{L_i} \log\left(\frac{1}{\beta_{l,i}}\right) - \sum_{l=1}^{L_i} \log\left(\frac{2}{\pi(1+\alpha_{l,i}^2)}\right) \\ & + \frac{1}{2} \sum_{l=1}^{L_i} \log\left(\frac{S^2}{\beta_{l,i}^2}\left[\alpha_{l,i}\psi^{(1)}(\alpha_{l,i}) - 1\right]\right) \\ & - \sum_{s=1}^{S} \log\left(\sum_{l=1}^{L_i} m_{l,i} g(d_s; \alpha_{l,i}, \beta_{l,i})\right) + \mathrm{C}, \end{aligned} \tag{A.1.16}$$

where C refers to the constants of the overall expression which are independent of $\Theta_i$.

## A.2 Message Length Minimisation Using the EM Algorithm

The minimisation of the total message length $I(\Theta_i, [d_s]_{s=1}^S)$ with respect to $\Theta_i$ is mathematically intractable since the model parameters cannot be expressed in closed form. To that end, auxiliary variables $[u_s]_{s=1}^S$ specifying the assignment of each data point $d_s$ to a mixture component are introduced. If $d_s$ is thought to be generated by the $l$th mixture component, then $u_s = l$. In this way, $I([d_s]_{s=1}^S \mid \Theta_i)$

is modified to

$$I([u_s, d_s]_{s=1}^S, \mid \Theta_i) = -\sum_{s=1}^{S} \log(p(u_s = l)g(d_s; \alpha_{l,i}, \beta_{l,i})), \tag{A.2.1}$$

and thus, the modified total message length expression is

$$\begin{aligned} I([u_s, d_s]_{s=1}^S, \Theta_i) = \frac{L_i - 1}{2}\log(S) - \frac{1}{2}\sum_{l=1}^{L_i}\log(m_{l,i}) - \log(L_i - 1)! \\ - \sum_{l=1}^{L_i}\log\left(\frac{1}{\beta_{l,i}}\right) - \sum_{l=1}^{L_i}\log\left(\frac{2}{\pi(1+\alpha_{l,i}^2)}\right) \\ + \frac{1}{2}\sum_{l=1}^{L_i}\log\left(\frac{S^2}{\beta_{l,i}^2}\left[\alpha_{l,i}\psi^{(1)}(\alpha_{l,i}) - 1\right]\right) \\ - \sum_{s=1}^{S}\log(p(u_s = l)g(d_s; \alpha_{l,i}, \beta_{l,i})) + \mathrm{C}. \end{aligned} \tag{A.2.2}$$

The problem is then to solve

$$\hat{\Theta}_i = \arg\min_{\Theta_i} I([u_s, d_s]_{s=1}^S, \Theta_i). \tag{A.2.3}$$

However, because $u_s$ is also unknown for all $s$, the minimisation has to be done iteratively via the EM algorithm. For the E-step, we need to derive the auxiliary function $\mathcal{Q}(\Theta_i, \Theta_i^{[n]})$, that is, the expectation of $-I([u_s, d_s]_{s=1}^S, \Theta_i)$ with respect to the posterior probability $r_{ls}^{[n]} = p([u_s]_{s=1}^S \mid [d_s]_{s=1}^S, \Theta_i^{[n]})$ where $\Theta_i^{[n]}$ is the model parameters obtained from the previous iteration or the $n$th iteration. Note that the negative of $I([u_s, d_s]_{s=1}^S, \Theta_i)$ is used since the maximisation operation is performed in the EM algorithm while the original problem in (A.2.3) is a minimisation problem.

Formally, the auxiliary function takes the form

$$\begin{aligned} \mathcal{Q}(\Theta_i, \Theta_i^{[n]}) &= E\left[-I([u_s, d_s]_{s=1}^S, \Theta_i) \mid [u_s]_{s=1}^S, \Theta_i^{[n]}\right] \\ &= -\sum_{u_1=1}^{L_i}\cdots\sum_{u_S=1}^{L_i} I([u_s, d_s]_{s=1}^S, \Theta_i)\prod_{s=1}^{S} r_{ls}^{[n]}, \end{aligned} \tag{A.2.4}$$

where

$$r_{ls}^{[n]} = \frac{m_{l,i}^{[n]} g(d_s; \alpha_{l,i}^{[n]}, \beta_{l,i}^{[n]})}{\sum_{h=1}^{L_i} m_{h,i}^{[n]} g(d_s; \alpha_{h,i}^{[n]}, \beta_{h,i}^{[n]})}. \tag{A.2.5}$$

As $I(\Theta_i)$ of $I([u_s, d_s]_{s=1}^S, \Theta_i)$ is independent of $[u_s]_{s=1}^S$, it can be pulled out of the summations, leaving only $I([u_s, d_s]_{s=1}^S \mid \Theta_i)$ inside. Also, we will omit $C$ from the subsequent derivations for the same reason and to prevent clutter. Resuming from before and simplifying, we get

$$\begin{aligned}\mathcal{Q}(\Theta_i, \Theta_i^{[n]}) = &-\frac{L_i - 1}{2}\log(S) + \frac{1}{2}\sum_{l=1}^{L_i}\log(m_{l,i}) + \log(L_i - 1)! \\ &+ \sum_{l=1}^{L_i}\log\left(\frac{1}{\beta_{l,i}}\right) + \sum_{l=1}^{L_i}\log\left(\frac{2}{\pi(1+\alpha_{l,i}^2)}\right) \\ &- \frac{1}{2}\sum_{l=1}^{L_i}\log\left(\frac{S^2}{\beta_{l,i}^2}\left[\alpha_{l,i}\psi^{(1)}(\alpha_{l,i}) - 1\right]\right) \\ &+ \sum_{l=1}^{L_i}\sum_{s=1}^{S}\log(m_{l,i})p(u_s = l \mid d_s, \Theta_i^{[n]}) \\ &+ \sum_{l=1}^{L_i}\sum_{s=1}^{S}\log(g(d_s; \alpha_{l,i}, \beta_{l,i}))p(u_s = l \mid d_s, \Theta_i^{[n]}).\end{aligned} \tag{A.2.6}$$

This concludes the E-step.

For the M-step, we first take the partial derivative of $Q(\Theta_i, \Theta_i^{[n]})$ with respect to $m_{l,i}$, $\alpha_{l,i}$ and $\beta_{l,i}$ before equating them separately to 0, i.e.

$$\frac{\partial\mathcal{Q}(\Theta_i, \Theta_i^{[n]})}{\partial m_{l,i}} = 0 \tag{A.2.7}$$

$$\frac{\partial\mathcal{Q}(\Theta_i, \Theta_i^{[n]})}{\partial \alpha_{l,i}} = 0 \tag{A.2.8}$$

$$\frac{\partial\mathcal{Q}(\Theta_i, \Theta_i^{[n]})}{\partial \beta_{l,i}} = 0. \tag{A.2.9}$$

After solving for their roots, we obtain

$$m_{l,i}^{[n+1]} = \frac{\frac{1}{2} + S_l^{[n]}}{S + \frac{L_i}{2}} \tag{A.2.10}$$

$$\beta_{l,i}^{[n+1]} = \frac{\sum_{s=1}^{S} d_s r_{ls}^{[n]}}{\alpha_{l,i}^{[n]} S_l^{[n]}}, \tag{A.2.11}$$

where $S_l^{[n]} = \sum_{s=1}^{S} r_{ls}^{[n]}$. The expression for $\alpha_{l,i}^{[n+1]}$ does not have a closed form, since

$$\begin{aligned}\log\left(\frac{\sum_{s=1}^{S} d_s r_{ls}^{[n]}}{S_l^{[n]}}\right) - \frac{\sum_{s=1}^{S} r_{ls}^{[n]} \log(d_s)}{S_l^{[n]}} + \frac{2\alpha_{l,i}^{[n+1]}}{S_l^{[n]}\left[1+\left(\alpha_{l,i}^{[n+1]}\right)^2\right]} \\ + \frac{1}{2S_l^{[n]}}\left(\frac{\alpha_{l,i}^{[n+1]}\psi^{(2)}(\alpha_{l,i}^{[n+1]}) + \psi^{(1)}(\alpha_{l,i}^{[n+1]})}{\alpha_{l,i}^{[n+1]}\psi^{(1)}(\alpha_{l,i}^{[n+1]}) - 1}\right) \\ - \log(\alpha_{l,i}^{[n+1]}) + \psi^{(0)}(\alpha_{l,i}^{[n+1]}) = 0.\end{aligned} \tag{A.2.12}$$

Therefore, $\alpha_{l,i}^{[n+1]}$ has to be searched for numerically using root-finding algorithms.

APPENDIX TWO

# PBDT TOOLKIT APPLICATION

This appendix presents the graphical application created as part of this research for facilitating the understanding of the estimates obtained using the PBDT algorithm.

Figure B.0.1 shows the main tab of the developed application. It allows the disaggregation algorithms and their various parameters to be configured. It also lets the user selects the appliances that should be extracted, among others. Other views in this tab are mostly for changing the display of the plots.

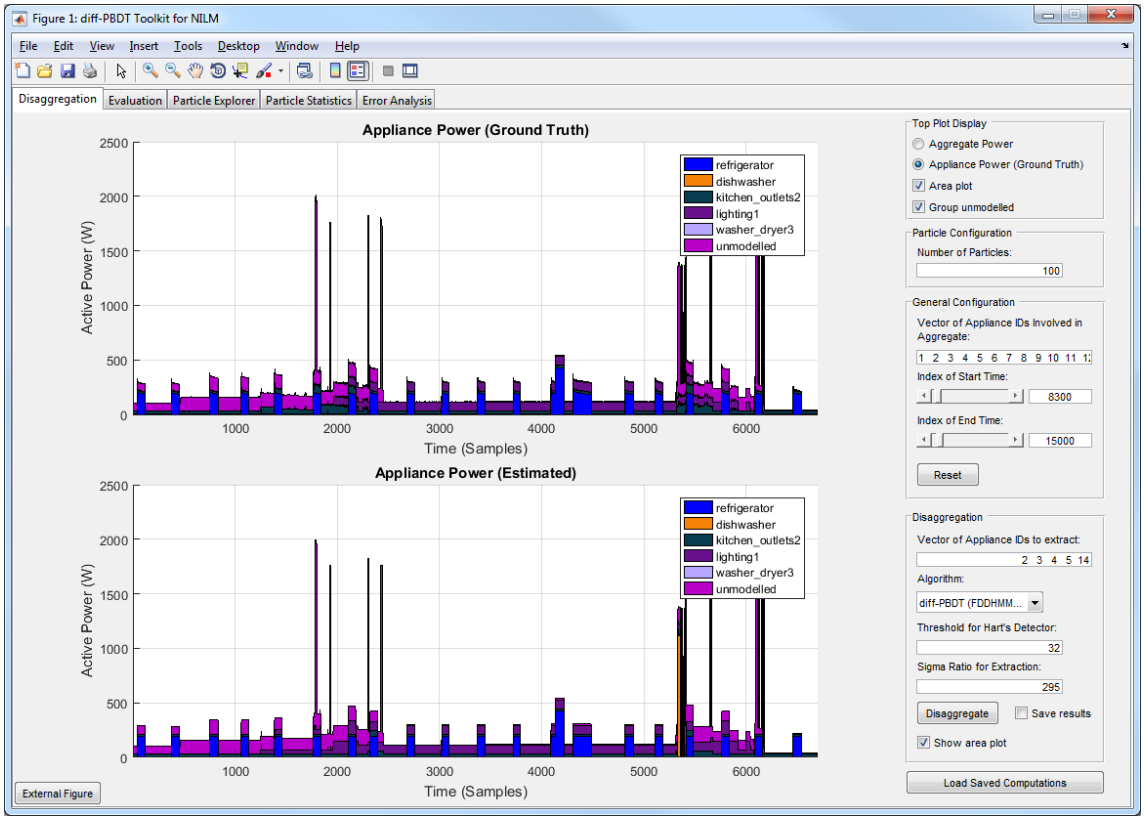



Figure B.0.1: The main tab of the developed GUI application.

The second tab of the application is shown in Figure B.0.2, in which accuracy of the disaggregation and the overall summary of the disaggregated energy are displayed.

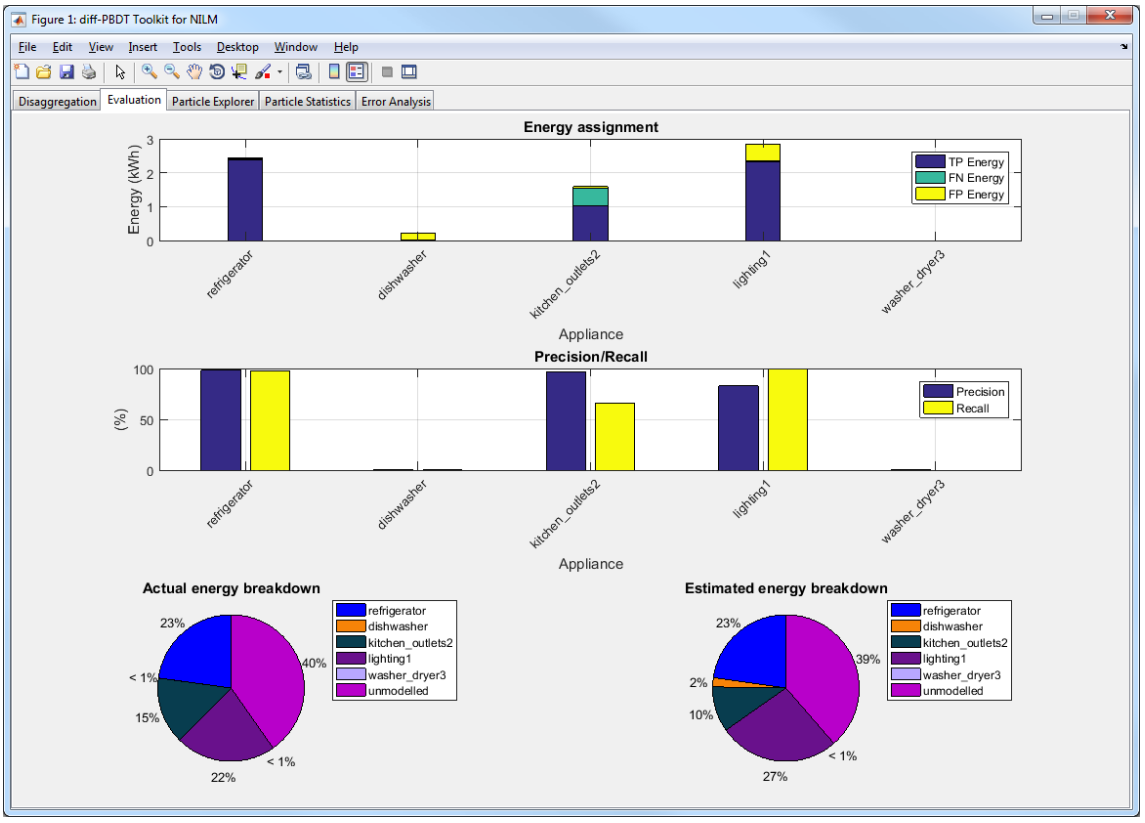



Figure B.0.2: The second tab of the developed GUI application.

The third tab of the application is presented in Figure B.0.3. Its main purpose is to display the estimates represented by different particles of the PBDT algorithm. This view enables wrong estimates to be investigated at a very detailed level and has been used as the basis for explaining the mistakes made by the algorithm in Chapter 4.

On the other hand, the fourth tab, as shown in Figure B.0.4, provides a view on each particle's ancestry, which also facilitates the task of identifying and understanding the failure modes of the algorithm.

Finally, Figure B.0.5 allows us to visualise the Cumulative Error Log Likehood Ratio (CELLR) explained in Chapter 4 for attributing whether disaggregation errors are due to the truncation procedure of PBDT or a result of inaccuracies in the emission model.

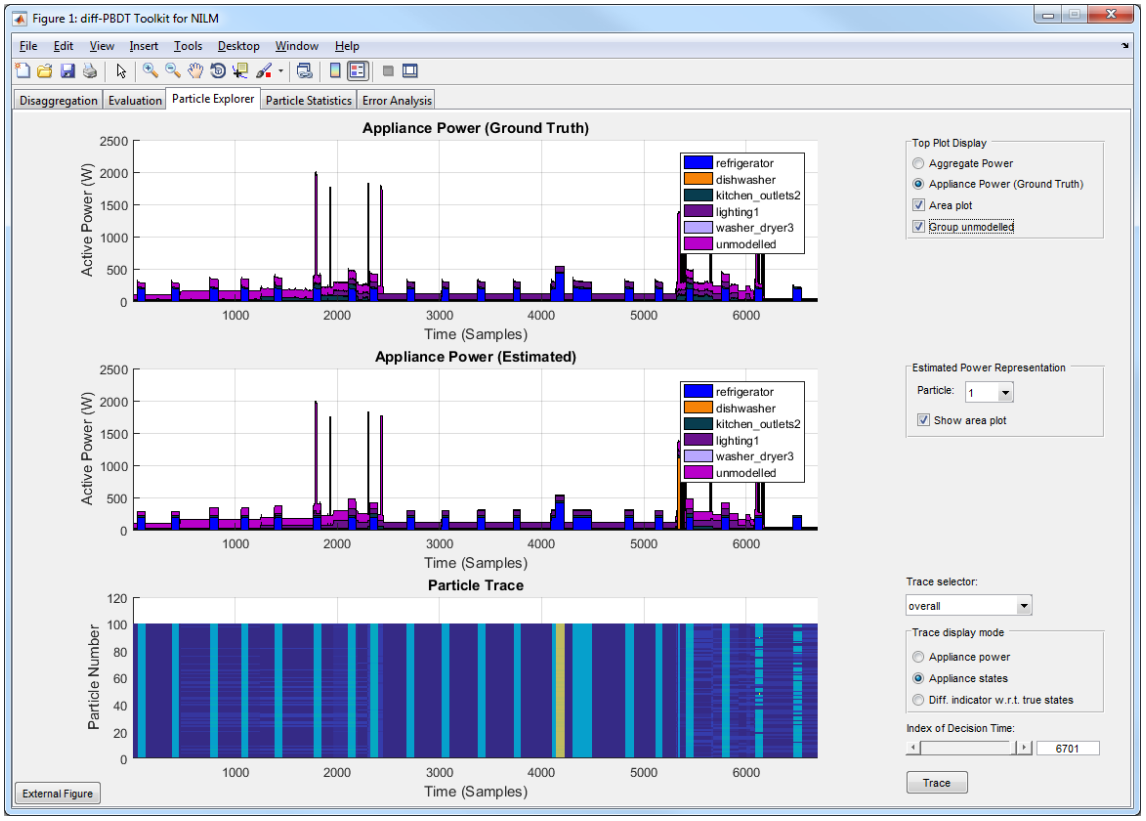



Figure B.0.3: The third tab of the developed GUI application.

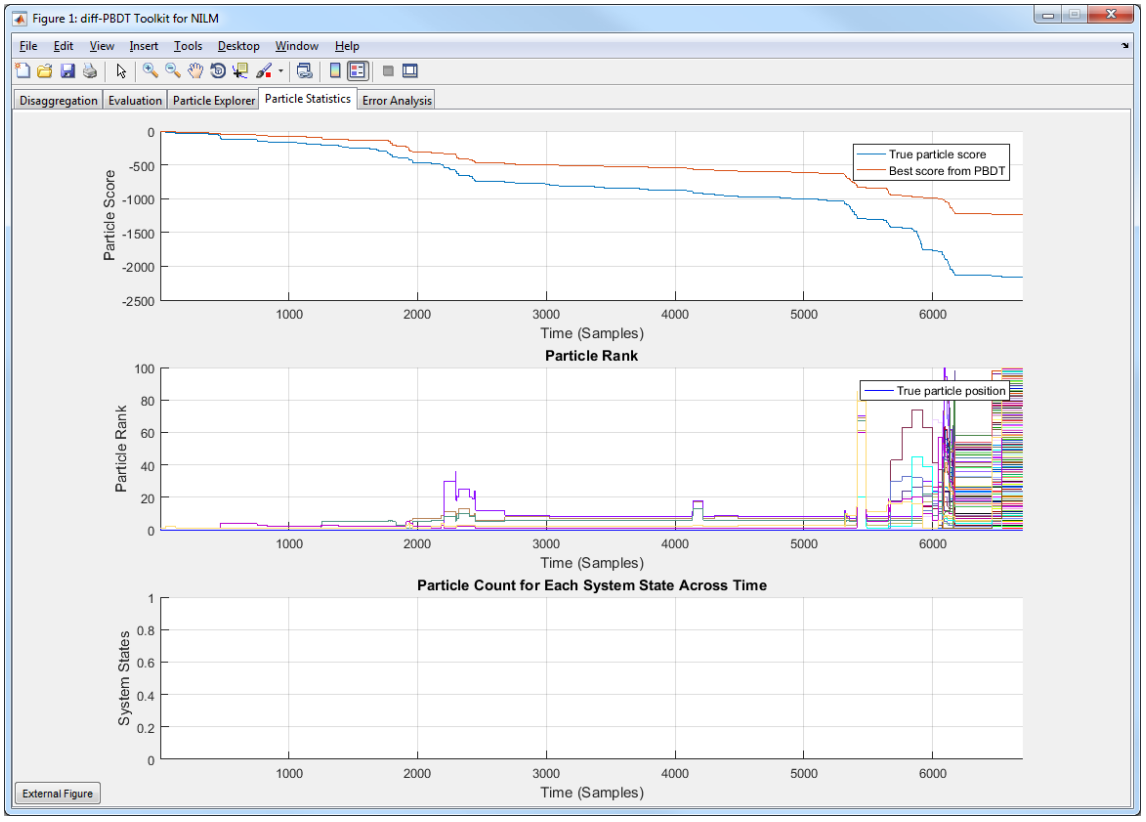



Figure B.0.4: The fourth tab of the developed GUI application.

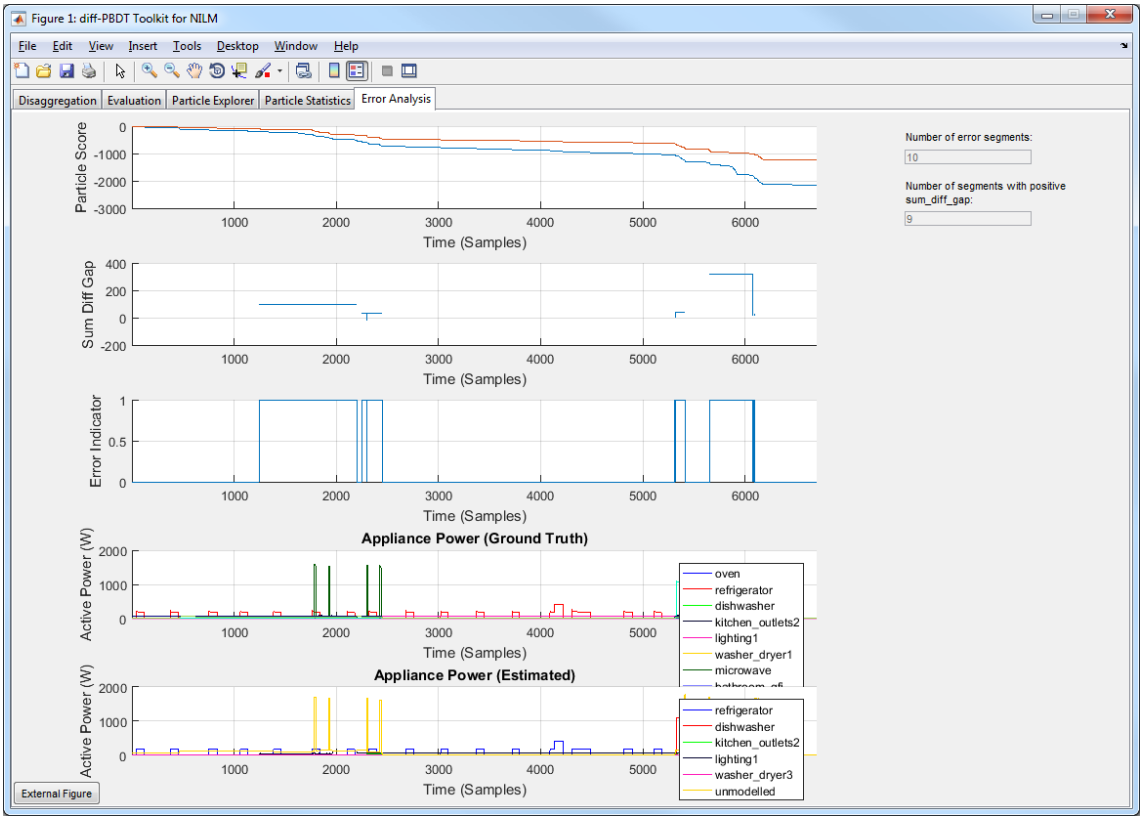



Figure B.0.5: The fifth tab of the developed GUI application.

# Bibliography


[AD03] Y. Agusta and D. L. Dowe. Unsupervised Learning of Gamma Mixture Models Using Minimum Message Length. *Proc. Third IASTED Conf. Artificial Intelligence and Applications*, (Mml):457–462, 2003.

[AM07] R. P. Adams and D. J. C. MacKay. Bayesian Online Changepoint Detection. Technical report, October 2007. `http://arxiv.org/abs/0710.3742`.

[AOB+12] K. Anderson, A. Ocneanu, D. Benitez, D. Carlson, A. Rowe, and M. Berges. BLUED: A Fully Labeled Public Dataset for Event-Based Non-Intrusive Load Monitoring Research. In *Proceedings of the 2nd Workshop on Data Mining Applications in Sustainability (SustKDD)*, Beijing, China, August 2012.

[App16] A. Appleby. SMHasher. `https://github.com/aappleby/smhasher`, 2016.

[BDS13] N. Batra, H. Dutta, and A. Singh. INDiC: Improved Non-intrusive Load Monitoring Using Load Division and Calibration. In *2013 12th International Conference on Machine Learning and Applications*, pages 79–84. IEEE, December 2013.

[Bel03] R. Bellman. *Dynamic Programming*. Courier Corporation, 2003.

[BGG94] M. Bellare, O. Goldreich, and S. Goldwasser. Incremental Cryptography: The Case of Hashing and Signing. In *Advances in Cryptology CRYPTO '94*, volume 839, pages 216–233. Springer Berlin Heidelberg, Berlin, Heidelberg, 1994.

[BGMS10] M. E. Berges, E. Goldman, H. S. Matthews, and L. Soibelman. Enhancing Electricity Audits in Residential Buildings with Nonintrusive Load Monitoring. *Journal of Industrial Ecology*, 14(5):844–858, October 2010.

[Bil98] J. Bilmes. A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov

Models. Technical report, 1998. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.28.613.

[BJJ+11] D. Bergman, D. Jin, J. Juen, N. Tanaka, C. Gunter, and A. Wright. Nonintrusive Load-Shed Verification. *IEEE Pervasive Computing*, 10(1):49–57, January 2011.

[BM97] M. Bellare and D. Micciancio. A New Paradigm for Collision-Free Hashing: Incrementality at Reduced Cost. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1233:163–192, 1997.

[BR08] J. Bloit and X. Rodet. Short-time Viterbi for Online HMM Decoding: Evaluation on a Real-Time Phone Recognition Task. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 2121–2124, 2008.

[BSM10] M. Bergés, L. Soibelman, and H. S. Matthews. Leveraging Data from Environmental Sensors to Enhance Electrical Load Disaggregation Algorithms. In *Proceeding of the 13th International Conference on Computing in Civil and Building Engineering*, Nottingham, UK, 2010.

[CA98a] A. I. Cole and A. Albicki. Algorithm for Nonintrusive Identification of Residential Appliances. In *Proceedings of the 1998 IEEE Internation Symposium on Circuits and Systems (ISCAS '98)*, volume 3, pages 338–341. IEEE, 1998.

[CA98b] A. I. Cole and A. Albicki. Data Extraction for Effective Non-Intrusive Identification of Residential Power Loads. In *Proceedings of the 1998 IEEE Instrumentation and Measurement Technology Conference. "Where Instrumentation is Going" (IMTC '98)*, volume 2, pages 812–815. IEEE, 1998.

[Car98] J. F. Cardoso. Blind Signal Separation: Statistical Principles. *Proceedings of the IEEE*, 86(10):2009–2025, 1998.

[CBS+13] D. Chen, S. Barker, A. Subbaswamy, D. Irwin, and P. Shenoy. Non-Intrusive Occupancy Monitoring using Smart Meters. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings - BuildSys'13*, pages 1–8, New York, New York, USA, 2013. ACM Press.

[CC13] P. Chou and R. Chang. Unsupervised Adaptive Non-intrusive Load Monitoring System. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pages 3180–3185. IEEE, October 2013.

[CL13] H. Chang and W. Lee. Energy Spectrum-Based Wavelet Rransform for Non-Intrusive Demand Monitoring and Load Identification. *2013 IEEE Industry Applications Society Annual Meeting*, pages 1–9, October 2013.

[CLSN06] R. Cox, S. B. Leeb, S. R. Shaw, and L. K. Norford. Transient Event Detection for Nonintrusive Load Monitoring and Demand Side Management Using Voltage Distortion. *Computer Engineering*, pages 1751–1757, 2006.

[CS84] J. H. Conway and N. Sloane. On the Voronoi Regions of Certain Lattices. *SIAM Journal on Algebraic Discrete Methods*, 5(3):294–305, 1984.

[CYL08] H. Chang, H. Yang, and C. Lin. Load Identification in Neural Networks for a Non-intrusive Monitoring of Industrial Electrical Loads. In *Proceedings of the 11th Internation Conference on Computer Supported Cooperative Work in Design 2007 (CSCWD '07)*, pages 664–674, 2008.

[DCL+05] T. DeNucci, R. Cox, S. B. Leeb, J. Paris, T. J. McCoy, C. Laughman, and W. C. Greene. Diagnostic Indicators for Shipboard Systems using Non-Intrusive Load Monitoring. In *IEEE Electric Ship Technologies Symposium, 2005.*, volume 2005, pages 413–420. IEEE, 2005.

[DJ08] A. Doucet and A. M. Johansen. A Tutorial on Particle Filtering and Smoothing : Fifteen Years Later. Technical Report December, 2008.

[DROS13] R. Dong, L. Ratliff, H. Ohlsson, and S. S. Sastry. A Dynamical Systems Approach to Energy Disaggregation. In *52nd IEEE Conference on Decision and Control*, April 2013.

[DWW12] M. Dewar, C. Wiggins, and F. Wood. Inference in Hidden Markov Models with Explicit State Duration Distributions. *IEEE Signal Processing Letters*, 19(4):235–238, April 2012.

[EBE13] D. Egarter, V. P. Bhuvana, and W. Elmenreich. Appliance State Estimation based on Particle Filtering. In *ACM BuildSys Workshop 2013*, pages 1–2, 2013.

[EBE15] D. Egarter, V. P. Bhuvana, and W. Elmenreich. PALDi: Online Load Disaggregation Via Particle Filtering. *IEEE Transactions on Instrumentation and Measurement*, 64(2):467–477, 2015.

[ES15] E. Elhamifar and S. Sastry. Energy Disaggregation via Learning 'Powerlets' and Sparse Coding. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 629–635. AAAI Press, 2015.

[FLC+09] J. E. Froehlich, E. Larson, T. Campbell, C. Haggerty, J. Fogarty, and S. N. Patel. HydroSense: Infrastructure-Mediated Single-Point Sensing of Whole-Home Water Activity. In *Proceedings of the 11th international conference on Ubiquitous computing - Ubicomp '09*, page 235, New York, New York, USA, 2009. ACM Press.

[FRA13] M. Figueiredo, B. Ribeiro, and A. D. Almeida. Electrical Signal Source Separation Via Nonnegative Tensor Factorization Using On Site Measurements in a Smart Home. *IEEE Transactions on Instrumentation and Measurement*, pages 1–10, 2013.

[GJ97] Z. Ghahramani and M. I. Jordan. Factorial Hidden Markov Models. *Machine Learning*, 29(2/3):245–273, 1997.

[GRP10] S. Gupta, M. S. Reynolds, and S. N. Patel. ElectriSense. In *Proceedings of the 12th ACM international conference on Ubiquitous computing - Ubicomp '10*, page 139, New York, New York, USA, 2010. ACM Press.

[GWK15] Z. Guo, Z. J. Wang, and A. Kashani. Home appliance Load Modeling from Aggregated Smart Meter Data. *IEEE Transactions on Power Systems*, 30(1):254–262, 2015.

[GY93] M. Gales and S. J. Young. *The Theory of Segmental Hidden Markov Models*. University of Cambridge, Department of Engineering, 1993.

[Har85] G. W. Hart. Prototype Nonintrusive Appliance Load Monitor. Technical report, MIT Energy Laboratory, 1985. `http://www.georgehart.com/research/Hart1985.pdf`.

[Har92] G.W. Hart. Nonintrusive Appliance Load Monitoring. *Proceedings of the IEEE*, 80(12):1870–1891, 1992.

[IS16] A. Imran and M. A. Syrus. An Improved Event Detection Algorithm for Non- Intrusive Load Monitoring System for Low Frequency Smart Meters. In *NILM Workshop 2016*, number 1, 2016.

[Jen05] S. P. Jenkins. Survival Analysis. *Unpublished Manuscript, Institute for Social and Economic Research, University of Essex, Colchester, UK*, 2005. `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.176.7572&rep=rep1&type=pdf`.

[Jer77] A. J. Jerri. The Shannon Sampling Theorem – Its Various Extensions and Applications: A Tutorial Review. *Proceedings of the IEEE*, 65(11):1565–1596, 1977.

[JLL+12] L. Jiang, J. Li, S. Luo, S. West, and G. Platt. Power Load Event Detection and Classification Based on Edge Symbol Analysis and Support Vector Machine. *Applied Computational Intelligence and Soft Computing*, 2012:1–10, 2012.

[Joh05] M. T. Johnson. Capacity and Complexity of HMM Duration Modeling Techniques. *IEEE Signal Processing Letters*, 12(5):407–410, May 2005.

[Jor95] M. I. Jordan. Why the Logistic Function? A Tutorial Discussion on Probabilities and Neural Networks, 1995. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.476.1842&rep=rep1&type=pdf.

[JR90] B. H. Juang and L. R. Rabiner. The Segmental K-means Algorithm for Estimating Parameters of Hidden Markov Models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(9):1639–1641, 1990.

[JW13] M. J. Johnson and A. S. Willsky. Bayesian Nonparametric Hidden Semi-Markov Models. *Journal of Machine Learning Research*, 14:673–701, February 2013.

[KA15] P. Kasarapu and L. Allison. Minimum Message Length Estimation of Mixtures of Multivariate Gaussian and von Mises-Fisher Distributions. *Machine Learning*, 100(2-3):333–378, 2015.

[KAL11] H. Kim, M. Arlitt, and G. Lyon. Unsupervised Disaggregation of Low Frequency Power Measurements. In *Proceedings of the Eleventh SIAM International Conference on Data Mining*, pages 747–758, 2011.

[KBN10] J. Z. Kolter, S. Batra, and A. Y. Ng. Energy Disaggregation via Discriminative Sparse Coding. In *Advances in Neural Information Processing Systems*, pages 1153–1161, 2010.

[KDH+16] W. Kong, Z. Y. Dong, D. J. Hill, f. Luo, and Y. Xu. Improving Non-Intrusive Load Monitoring Efficiency via a Hybrid Programming Method. *IEEE Transactions on Industrial Informatics*, 3203(c):1–1, 2016.

[KDM+16] W. Kong, Z. Y. Dong, J. Ma, D. Hill, J. Zhao, and F. Luo. An Extensible Approach for Non-Intrusive Load Disaggregation with Smart Meter Data. *IEEE Transactions on Smart Grid*, 3053(c):1–1, 2016.

[KJ11] J. Z. Kolter and M. J. Johnson. REDD : A Public Data Set for Energy Disaggregation Research. In *SustKDD workshop on Data Mining Applications in Sustainability*, number 1, pages 1–6, 2011.

[KJ12] J. Z. Kolter and T. Jaakkola. Approximate Inference in Additive Factorial HMMs with Application to Energy Disaggregation. In *2012 International Conference on Artificial Intelligence and Statistics*, pages 1472–1482, 2012.

[KK15a] J. Kelly and W. Knottenbelt. Neural NILM: Deep Neural Networks Applied to Energy Disaggregation. In *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments - BuildSys '15*, pages 55–64, New York, New York, USA, 2015. ACM Press.

[KK15b] J. Kelly and W. Knottenbelt. The UK-DALE Dataset, Domestic Appliance-level Electricity Demand and Whole-house Demand from Five UK Homes. *Scientific Data*, 2:150007, March 2015.

[KKP06] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas. Handling Imbalanced Datasets: A Review. *GESTS International Transactions on Computer Science and Engineering*, 30(1):25–36, 2006.

[KLL97] U. A. Khan, S. B. Leeb, and M. C. Lee. A Multiprocessor for Transient Event Detection. *IEEE Transactions on Power Delivery*, 12(1):51–60, 1997.

[KM91] V. Krishnamurthy and J. B. Moore. Signal Processing of Semi-Markov Models with Exponentially Decaying States. In *Decision and Control, 1991., Proceedings of the 30th IEEE Conference on*, pages 2744–2749. IEEE, 1991.

[Knu92] D. E Knuth. Two Notes on Notation. *The American Mathematical Monthly*, 99(5):403, May 1992.

[Kot07] S. B. Kotsiantis. Supervised machine learning: A review of classification techniques. *Informatica*, 31(3):249–269, 2007.

[KX03] D. Karlis and E. Xekalaki. Choosing Initial Values for the EM Algorithm for Finite Mixtures. *Computational Statistics & Data Analysis*, 41(3-4):577–590, January 2003.

[KZZS13] Z. Kang, Y. Zhou, L. Zhang, and C. J. Spanos. Virtual Power Sensing Based on a Multiple-Hypothesis Sequential Test. In *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 785–790, 2013.

[LB16] Henning Lange and Mario Bergés. Efficient Inference in Dual-emission FHMM for Energy Disaggregation. In *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[LFL07] H. Lam, G. Fung, and W. Lee. A Novel Method to Construct Taxonomy Electrical Appliances Based on Load Signaturesof. *IEEE Transactions on Consumer Electronics*, 53(2):653–660, 2007.

[LKC+03] C. Laughman, L. Kwangduk, R. Cox, S. Shaw, S. Leeb, L. Norford, and P. Armstrong. Power Signature Analysis. *IEEE Power and Energy Magazine*, 1(2):56–63, March 2003.

[LKLS93] S. B. Leeb, J. L. Kirtley, M. S. Levan, and J. P. Sweeney. Development and Validation of a Transient Event Detector. *AMP Journal of Technology*, 3:69–74, 1993.

[LNKC10a] J. Liang, S. Ng, G. Kendall, and J. Cheng. Load Signature Study–Part I: Basic Concept, Structure, and Methodology. *IEEE Transactions on Power Delivery*, 25(2):551–560, April 2010.

[LNKC10b] J. Liang, S. Ng, G. Kendall, and J. Cheng. Load Signature Study–Part II: Disaggregation Framework, Simulation, and Applications. *IEEE Transactions on Power Delivery*, 25(2):561–569, April 2010.

[LS99] D. D. Lee and H. S. Seung. Learning the Parts of Objects by Non-negative Matrix Factorization. *Nature*, 401(6755):788–91, October 1999.

[LSK95] S. B. Leeb, S. R. Shaw, and J. L. Kirtley. Transient Event Detection in Spectral Envelope Estimates. *Power*, 10(3):1200–1210, 1995.

[Luc97] A. Lucas. Robustness of the Student-t based M-estimator. *Communications in Statistics - Theory and Methods*, 26(5):1165–1182, January 1997.

[LWP12] J. Li, S. West, and G. Platt. Power Decomposition Based on SVM Regression. In *2012 Proceedings of International Conference on Modelling, Identification and Control*, pages 1195 – 1199, 2012.

[MHHE11] A. Marchiori, D. Hakkarinen, Q. Han, and L. Earle. Circuit-Level Load Monitoring for Household Energy Management. *IEEE Pervasive Computing*, 10(1):40–48, January 2011.

[MPB+13] S. Makonin, F. Popowich, L. Bartram, B. Gill, and I. V. Bajic. AMPds: A Public Dataset for Load Disaggregation and Eco-Feedback Research. In *2013 IEEE Electrical Power & Energy Conference*, number EPEC, pages 1–6. IEEE, August 2013.

[MPB+16] S. Makonin, F. Popowich, I. V. Bajic, B. Gill, and L. Bartram. Exploiting HMM Sparsity to Perform Online Real-Time Nonintrusive Load Monitoring. *IEEE Transactions on Smart Grid*, 7(6):2575–2585, 2016.

[NJ02] A. Y. Ng and M. I. Jordan. On Discriminative Vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 841–848. MIT Press, 2002.

[NL96] L. K. Norford and S. B. Leeb. Non-intrusive Electrical Load Monitoring in Commercial Buildings Based on Steady-State and Transient Load-Detection Algorithms. *Energy and Buildings*, 24(1):51–64, January 1996.

[NSM11] NSMP Business Requirements Work Group. Smart Metering Infrastructure Minimum Functionality Specification. Technical Report May, 2011.

[OBW96] J. Oliver, R. Baster, and C. Wallace. Unsupervised Learning Using MML. In *In Machine Learning: Proceedings of the Thirteenth International Conference (ICML 96*, pages 364–372. Morgan Kaufmann Publishers, 1996.

[PGWR12] O. Parson, S. Ghosh, M. Weal, and A. Rogers. Non-Intrusive Load Monitoring Using Prior Models of General Appliance Types. In *AAAI*, 2012.

[PGWR14] O. Parson, S. Ghosh, M. Weal, and A. Rogers. An Unsupervised Training Method for Non-intrusive Appliance Load Monitoring. *Artificial Intelligence*, 217:1–19, December 2014.

[PM00] D. Peel and G. J. McLachlan. Robust Mixture Modelling using the t-distribution. *Statistics and Computing*, 10:339–348, 2000.

[PRK+07] S. N. Patel, T. Robertson, J. A. Kientz, M. S. Reynolds, and G. D. Abowd. At the Flick of a Switch: Detecting and Classifying Unique Electrical Events on the Residential Power Line (Nominated for the Best Paper Award). In *UbiComp 2007: Ubiquitous Computing*, pages 271–288. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[PSJ+07] J. E. Petersen, V. Shunturov, K. Janda, G. Platt, and K. Weinberger. Dormitory Residents Reduce Electricity Consumption When Exposed to RealTime Visual Feedback and Incentives. *International Journal of Sustainability in Higher Education*, 8(1):16–33, January 2007.

[Ras04] C. E. Rasmussen. Gaussian Processes in Machine Learning. In *International Journal of Neural Systems*, volume 14, pages 63–71. April 2004.

[RCG12] S. Rahimi, A. D. C. Chan, and R. A. Goubran. Nonintrusive Load Monitoring of Electrical Devices in Health Smart Homes. In *2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, pages 2313–2316. IEEE, May 2012.

[RLBH94] J. G. Roos, I. E. Lane, E. C. Botha, and G. P. Hancke. Using Neural Networks for Non-intrusive Monitoring of Industrial Electrical Loads. In *Proceedings of the 10th IEEE Instrumentaion and Measurement Technology Conference 1994 (IMTC '94)*, pages 1115–1118. IEEE, 1994.

[RNSO10] A. G. Ruzzelli, C. Nicolas, A. Schoofs, and G. M. P. O'Hare. Real-Time Recognition and Profiling of Appliances through a Single Electricity Sensor. In *2010 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 1–9. IEEE, June 2010.

[Rus93] M. Russell. A Segmental HMM for Speech Pattern Modelling. In *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, volume 2, pages 499–502. IEEE, 1993.

[RW92] P. Ramesh and J. G. Wilpon. Modeling State Durations in Hidden Markov Mdels for Automatic Speech Recognition. In *Proceedings ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, number i, pages 381–384 vol.1. IEEE, 1992.

[SDF04] A. Shahrokni, T. Drummond, and P. Fua. Texture Boundary Detection for Real-Time Tracking. In *In European Conference on Computer Vision*, pages 566–577, 2004.

[SIS+08] K. Suzuki, S. Inagaki, T. Suzuki, H. Nakamura, and K. Ito. Nonintrusive Appliance Load Monitoring Based on Integer Programming. In *2008 SICE Annual Conference*, pages 2742–2747. IEEE, August 2008.

[SLNC08] S. R. Shaw, S. B. Leeb, L. K. Norford, and R. W. Cox. Nonintrusive Load Monitoring and Diagnostics in Power Systems. *IEEE Transactions on Instrumentation and Measurement*, 57(7):1445–1454, July 2008.

[SLS14] V. Stankovic, J. Liao, and L. Stankovic. A Graph-Based Signal Processing Approach for Low-Rate Energy Disaggregation. In *2014 IEEE Symposium on Computational Intelligence for Engineering Solutions (CIES)*, pages 81–87. IEEE, December 2014.

[SNL06] D. Srinivasan, W. S. Ng, and A. C. Liew. Neural-Network-Based Signature Recognition for Harmonic Source Identification. *IEEE Transactions on Power Delivery*, 21(1):398–405, 2006.

[SNS15] M. Sundermeyer, H. Ney, and R. Schluter. From Feedforward to Recurrent LSTM Neural Networks for Language Modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):517–529, March 2015.

[SU09] L. G. Swan and V. I. Ugursal. Modeling of end-use Energy Consumption in the Residential Sector: A Review of Modeling Techniques. *Renewable and Sustainable Energy Reviews*, 13(8):1819–1835, oct 2009.

[Sul91] F. Sultanem. Using Appliance Signatures for Monitoring Residential Loads at Meter Panel Level. *IEEE Transactions on Power Delivery*, 6(4):1380–1385, 1991.

[SY12] R. Streubel and B. Yang. Identification of Electrical Appliances via Analysis of Power Consumption. In *2012 47th International Universities Power Engineering Conference (UPEC)*, pages 1–6. IEEE, September 2012.

[TWLT16] G. Tang, K. Wu, J. Lei, and J. Tang. A Simple and Robust Approach to Energy Disaggregation in the Presence of Outliers. *Sustainable Computing: Informatics and Systems*, 9:8–19, 2016.

[TWLX15] G. Tang, K. Wu, J. Lei, and W. Xiao. The Meter Tells You Are At Home! Non-intrusive Occupancy Detection via Load Curve Data. In *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 897–902. IEEE, November 2015.

[U.S16] U.S. Energy Information Administration. Electricity. In *International Energy Outlook 2016*, chapter 5, pages 83–84. May 2016. `http://www.eia.gov/forecasts/ieo/pdf/0484(2016).pdf`.

[Vas91] S. V. Vaseghi. Hidden Markov Models with Duration-Dependent State Transition Probabilities. *Electronics Letters*, 27(8):625, 1991.

[WB68] C. S. Wallace and D. M. Boulton. An Information Measure for Classification. *The Computer Journal*, 11(2):185–194, August 1968.

[WcD14] Y. F. Wong, Y. A. Şekercioğlu, and T. Drummond. Real-time Load Disaggregation Algorithm using Particle-Based Distribution Truncation with State Occupancy Model. *Electronics Letters*, 50(9):697–699, 2014.

[WcDW13] Y. F. Wong, Y. A. Şekercioğlu, T. Drummond, and V. S. Wong. Recent Approaches to Non-intrusive Load Monitoring Techniques in Residential Settings. *IEEE Symposium on Computational Intelligence Applications in Smart Grid, CIASG*, pages 73–79, 2013.

[WF87] C. S. Wallace and P. R. Freeman. Estimation and Inference by Compact Coding. *Journal of the Royal Statistical Society*, 49(3):240–265, 1987.

[WJNB95] P. R. Wilson, M. S. Johnstone, M. Neely, and D. Boles. Dynamic Storage Allocation: A Survey and Critical Review. In *Memory Management*, pages 1–116. Springer, 1995.

[Wu83] C. F. J. Wu. On the Convergence Properties of the EM Algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.

[WWDc13] V. S. Wong, Y. F. Wong, T. Drummond, and Y. A. Şekercioğlu. A Fast Multiple Appliance Detection Algorithm for Non-Intrusive Load Monitoring. *IEEE Symposium on Computational Intelligence Applications in Smart Grid, CIASG*, pages 80–86, 2013.

[WZ12] Z. Wang and G. Zheng. Residential Appliances Identification and Monitoring by a Nonintrusive Method. *IEEE Transactions on Smart Grid*, 3(1):80–92, March 2012.

[Yu10] S. Yu. Hidden Semi-Markov Models. *Artificial Intelligence*, 174(2):215–243, February 2010.

[ZBZ11] T. Zia, D. Bruckner, and A. Zaidi. A Hidden Markov Model Based Procedure for Identifying Household Electric Loads. In *IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society*, pages 3218–3223. IEEE, November 2011.

[Zei12] M. Zeifman. Disaggregation of Home Energy Display Data using Probabilistic Approach. *IEEE Transactions on Consumer Electronics*, 58(1):23–31, February 2012.

[ZGIR12] A. Zoha, A. Gluhak, M. A. Imran, and S. Rajasegarar. Non-intrusive Load Monitoring Approaches for Disaggregated Energy Sensing: A Survey. *Sensors (Basel, Switzerland)*, 12(12):16838–66, January 2012.

[ZGN+12] A. Zoha, A. Gluhak, M. Nati, M. A. Imran, and S. Rajasegarar. Acoustic and Device Feature Fusion for Load Recognition. In *Intelligent Systems (IS), 2012 6th IEEE International Conference*, pages 386–392. IEEE, 2012.

[ZR11a] M. Zeifman and K. Roth. Nonintrusive Appliance Load Monitoring: Review and Outlook. *IEEE Transactions on Consumer Electronics*, 57(1):76–84, February 2011.

[ZR11b] M. Zeifman and K. Roth. Viterbi Algorithm with Sparse Transitions (VAST) for Nonintrusive Load Monitoring. In *Proceedings of the 2011 IEEE Symposium on Computational Intelligence Appliancations in Smart Grid (CIASG '11)*, pages 1–8. IEEE, April 2011.