UNIVERSITY OF SOUTHAMPTON

Algorithms for Appliance Usage Prediction

by

Ngoc Cuong Truong

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Faculty of Physical Sciences and Engineering
Electronics and Computer Science

June 2014

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF PHYSICAL SCIENCES AND ENGINEERING
ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

Algorithms for Appliance Usage Prediction

by Ngoc Cuong Truong

Demand-Side Management (DSM) is one of the key elements of future Smart Electricity Grids. DSM involves mechanisms to reduce or shift the consumption of electricity in an attempt to minimise peaks. By so doing it is possible to avoid using expensive peaking plants that are also highly carbon emitting. A key challenge in DSM, however, is the need to predict energy usage from specific home appliances accurately so that consumers can be notified to shift or reduce the use of high energy-consuming appliances. In some cases, such notifications may be also need to be given at very short notice. Hence, to solve the appliance usage prediction problem, in this thesis we develop novel algorithms that take into account both users' daily practices (by taking advantage of the cyclic nature of routine activities) and the inter–dependency between the usage of multiple appliances (i.e., the user's typical consumption patterns). We propose two prediction algorithms to satisfy the needs for fast prediction and high accuracy respectively: i) a rule-based approach, EGH-H, for scenarios in which notifications need to be given at short notice, to find significant patterns in the use of appliances that can capture the user's behaviour (or habits), ii) a graphical–model based approach, GM-PMA (Graphical Model for Prediction in Multiple Appliances) for scenarios that require high prediction accuracy. We demonstrate through extensive empirical evaluations on real–world data from a prominent database of home energy usage that GM-PMA outperforms existing methods by up to 41%, and the runtime of EGH-H is 100 times lower on average, than that of other benchmark algorithms, while maintaining competitive prediction accuracy. Moreover, we demonstrate the use of appliance usage prediction algorithms in the context of demand–side management by proposing an Intelligent Demand Responses (IDR) mechanism, where an agent uses Logistic Inference to learn the user's preferences, and hence provides the best *personalised* suggestions to the user. We use simulations to evaluate IDR on a number of user types, and show that, by using IDR, users are likely to improve their savings significantly.

Contents

Declaration of Authorship **xi**

Nomenclature **xiii**

Acronyms **xv**

1 Introduction **1**
- 1.1 Agent-based Demand Side Management 3
- 1.2 The Appliance Usage Prediction Problem 6
- 1.3 Research Requirements 7
- 1.4 Research Contributions 8
- 1.5 Report Outline 10

2 Literature Review **13**
- 2.1 The Smart Grid 13
  - 2.1.1 Smart Meters 15
  - 2.1.2 Demand-Side Management 17
- 2.2 Prediction with Inter–Dependent Data 19
  - 2.2.1 Temporal Point Processes 19
  - 2.2.2 Mining Dependencies in Temporal Event Sequences 20
    - 2.2.2.1 Frequent Episodes in Event Sequences 21
    - 2.2.2.2 Identify Frequent Episodes 21
    - 2.2.2.3 Frequent Episode Discovery for Prediction in Temporal Event Sequences 22
    - 2.2.2.4 The EGH algorithm 23
- 2.3 Predicting Human Routine 24
- 2.4 Probabilistic Graphical Models 25
  - 2.4.1 Hidden Markov Models 26
  - 2.4.2 Approximate Inference 27
    - 2.4.2.1 Gibbs Sampling 27
    - 2.4.2.2 Variational Bayes 28
- 2.5 Summary 28

3 Appliance Usage Prediction Model and Datasets **31**
- 3.1 Model Description 31
- 3.2 Real–world Datasets 32
  - 3.2.1 The REDD dataset 32
  - 3.2.2 The FigureEnergy Dataset 33

3.3 Summary 34

4 The EGH-H Algorithm 37
4.1 The Inter–Dependency Model 37
4.2 The Human Routine Model 38
4.3 The Mixture Model 39
4.4 Appliance Usage Prediction 40
4.5 Summary 41

5 The GM-PMA Algorithm 43
5.1 The Inter–Dependency Clustering Model 43
5.1.1 Likelihood Functions 44
5.1.2 Dirichlet Process Mixture 46
5.2 Inference of Parameters and Cluster Number 47
5.3 Appliance Usage Prediction 48
5.4 Summary 50

6 Empirical Evaluation 51
6.1 Experiment Settings 51
6.2 Benchmark Algorithms 51
6.3 Evaluation Metrics 52
6.4 Synthetic Data Evaluation 53
6.5 Empirical Evaluation on Real–World Datasets 55
6.5.1 Experiment 1: Performance on the REDD Dataset 56
6.5.2 Experiment 2: Performance on the FigureEnergy Dataset 56
6.5.3 Experiment 3: Human Routine Cycles 58
6.5.4 Experiment 4: Average Runtime 59
6.6 Summary 60

7 Intelligent Demand Response 61
7.1 The IDR Mechanism 61
7.2 The Human Response Model 62
7.2.1 User Utility Function 63
7.2.2 Decision–making given suggestions 64
7.3 Suggestion Optimisation 65
7.4 Learning from User Responses 66
7.5 Empirical Evaluation 67
7.5.1 User Profiles 67
7.5.2 Experiment 1: Evaluating IDR 69
7.5.3 Experiment 2: Evaluating Personalised Suggestions 70
7.6 Summary 74

8 Conclusions 77
8.1 Theoretical Implications 79
8.2 Practical Implications 80

References 81

List of Figures

1.1 An example of smart home model. (source: http://www.digikey.com) . . 4
1.2 An example of a display of real-time carbon intensity in the UK (source: http://realtimecarbon.org/). 5

2.1 A 'Liberty' smart meter manufactured by Secure Meters UK Ltd. 15
2.2 The AlertMe's meter reading kits. 16
2.3 A HMM represented as a graphical model. The left-to-right spatial dimension represents time. The output nodes Y_i are observation nodes, and the state nodes X_i are hidden. 26

3.1 An example of using annotation in the FE application. The graph denotes power consumptions and the highlighted segments are the annotations of the user. 33
3.2 An example of how to convert raw data of power consumption into a list of cyclic on–off events. 34

5.1 Graphical model for the usage of multi appliances in the home. Shaded nodes indicate observed information. 45
5.2 An example of the washing machine usage throughout a day. 46

6.1 Synthetic data evaluation for two appliances: washing machine and tumble dryer. The number of usage of washing machine is arbitrarily fixed at 3. The noise of inter-dependency between the appliances is varied. Note that the higher the noise of inter-dependency value are generated, the lower the level of inter-dependency between the two appliances is likely to achieve. 55
6.2 Synthetic data evaluation for two appliances: washing machine and tumble dryer. The level of inter-dependency between these appliances is fixed. 55
6.3 ROC curve of the algorithms run on house 1 - REDD. 56
6.4 ROC curve of the algorithms run on house 3 - REDD. 57
6.5 ROC curve of the algorithms run on house 4 - REDD. 57
6.6 ROC curve of the algorithms run on User FE22 from FigureEnergy. . . . 58
6.7 ROC curve of the algorithms run on User FE39 from FigureEnergy. . . . 58
6.8 Cyclic human routine evaluation on EGH-H for REDD (House 1) dataset. 59

7.1 Example of using logistic inference to learn the user's preferences from the user's responses. The algorithm finds a new gradient (the dashed line) to minimise the errors of agent's suggestions. Intuitively, the user would say 'yes' to the suggestion that the trade-off located above the line, and 'no' otherwise. 66

7.2 The 5 canonical ground truth user preferences we use in our experiments, represented as lines of different colour (representing the linear trade-off between savings and comfort). For savings-comfort values *above* each line, the rational response (i.e., without noise) for the simulated user is to accept the proposed appliance usage shift. Otherwise, the suggestion is rejected. However, since our model is probabilistic, the user may behave irrationally. The extent of irrationality is best understood as new parameter α under a change of basis for the utility function $U(s, t_i, t_j) = \frac{1}{\alpha}(w_c|t_j - t_i| + s)$. The 4 subfigures show different values for irrationality. The shaded areas indicate the 95% confidence region that the simulated response behaviour will be irrational (i.e., be the opposite response to the one dictated purely by utility theory). In our experiments, we fix $\alpha = 1$ (equivalent to $w_s = 1$ in the original basis given in Equation 7.2) as a reasonable choice of irrationality, resulting in the need to change only one parameter (w_c) throughout. 68
7.3 Learning user's behaviour error at House 1 (REDD dataset). 71
7.4 Learning user's behaviour error at House 3 (REDD dataset). 71
7.5 Learning user's behaviour error at House 4 (REDD dataset). 72
7.6 Potential savings by selecting the best personalised suggestions within House 1 (REDD dataset). 73
7.7 Potential savings by selecting the best personalised suggestions within House 3 (REDD dataset). 73
7.8 Potential savings by selecting the best personalised suggestions within House 4 (REDD dataset). 74

List of Tables

1.1 An overview of our contributions in terms of the research requirements in the appliance usage prediction problem. The symbols have the following meaning: '+' ('++') means that the requirement is (strongly) satisfied. In addition, '(*)' indicates the best performance of the row. On the other hand, '-' means the requirement is not satisfied. 10

3.1 An example of data collected from FigureEnergy for user 32. 35

6.1 Runtime performance for 8 appliances (in seconds). 59

7.1 Different User settings for different decision making. 69
7.2 Savings (in UK pence) achieved by different simulated human response behaviours performed on REDD dataset. 70

Declaration of Authorship

I, Ngoc Cuong Truong, declare that the thesis entitled *Algorithms for Appliance Usage Prediction* and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- parts of this work have been published as: (Truong et al., 2014), (Truong et al., 2013a), (Truong et al., 2013c), (Truong et al., 2013b), and (Truong et al., 2012).

Signed:...

Date:..

Nomenclature

$l \in 1, \ldots, L$	An appliance
$t \in 1, \ldots, T$	A time slice.
$n \in 1, \ldots, N$	The n^{th} day.
$x_{n,l,t} \in \{0, 1\}$	Indicating whether appliance l is being used at time t on day n.
$x_{n,l} = \langle x_{n,l,1}, \ldots, x_{n,l,T} \rangle$	Observations of appliance l on the n^{th} day.
$\mathbf{X} = \{x_{1,l}, \ldots, x_{n-1,l}\}$	Past observations of appliance usage.
φ	An episode.
η_φ	A noise parameter.
Δ_φ	Vector of appliance usage.
S	A state space.
T_φ	A total number of appliance usage in the training dataset.
f_φ	A frequency that the episode φ occurs in the training dataset.
$\hat{M}$	An episode's size.
$p(\varphi)$	A number of appliance usage within φ.
D	A training dataset.
$F^s = \{\varphi_1, \ldots, \varphi_J\}$	A set of significant episodes.
Λ_φ	An HMM for the episode φ.
Λ_l	A mixture model of HMMs for appliance l.
$\Theta^g = \{\theta_1^g, \ldots, \theta_J^g\}$	Mixture coefficients.
q_n	A day of the week.
C	Numbers of appliance l are being used in the training dataset.
$k \in K$	A number of day types.
z_n	An indicator parameter.
$\mu_{k,l,t} \in [0, 1]$	The probability of the appliance l belongs to the day type class k in the time of the day t.
$\mu_k = (\mu_{k,l,1} \ldots \mu_{k,l,T})$	A set of parameters of a day type.
$\mathcal{B}(\beta_1, \beta_2)$	Beta distribution with shape parameters β_1, β_2.
$\mathbf{M} = (\mu_1, \ldots, \mu_K)$	A set of data types.
$\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$	A set of day type indicators.
$\mathbf{V} = \{v_1, \ldots, v_K\}$	A set of weight parameters.
$\mathcal{M}(.)$	A multinomial distribution.
$\mathrm{Dir}(\omega)$	A Dirichlet distribution with parameter ω.

f_n: $U(s, t_i, t_j)$ An utility function.

Acronyms

DSM	Demand-Side Management
ADSM	Agent-based Demand-Side Management
EGH	Episode Generation Hidden Markov Model
EGH-H	Episode Generation Hidden Markov Model for Human
GM-PMA	Graphical Model for Predicting Multi-Appliances
DP	Dirichlet Process
DPM	Dirichlet Process Mixture
IHD	In-Home Display
DLC	Direct Load Control
SPM	Smart Pricing Mechanism
TOU	Time-Of-Use
CPP	Critical-Peak Pricing
RTP	Real-Time Pricing
PP	Poison Process
PNM	Poison Network Model
PCIM	Piecewise-Constant Conditional Intensity Model
FED	Frequent Episodes Discovery
HMM	Hidden Markov Model
MCMC	Markov chain Monte Carlo
REDD	Reference Energy Disaggregation Data set
FE	FigureEnergy
ROC	Receiver Operating Characteristic
TPR	True Positive Rate
FPR	False Positive Rate
FN	False Negative
TN	True Negative
FP	False Positive
TP	True Positive
SD	Synthetic Dataset
AUC	Area Under the Curve
IDR	Intelligent Demand Response
EM	Expectation Maximisation

Acknowledgements

I would like to thank my supervisors Sarvapali (Gopal) Ramchurn and Enrico Costanza for their invaluable support, encouragement and guidance throughout my entire PhD. Particularly, I am incredibly thankful for their time and patience in correcting my writing, and for their enlightening discussions that brought me closer to the solution.

I would like to thank my internal examiner Alex Rogers and my external examiner Alessandro Farinelli for letting my defense be an enjoyable moment, and for your brilliant comments and suggestions.

To my co-authors, with whom I have spent wonderful times of (not only) working: Long Tran-Thanh for the great help of keeping me on track with the direction of my PhD, James McInerney for providing much knowledge about inference and modelling. Their advices have been an important influence on my work.

I would like to thank my colleagues and friends in the Agents, Interaction, and Complexity Research group (specially the agents part of the group). It has been pleasure to collaborate and socialise with such a great bunch of people.

To my parents, my brother and my family, for their unconditional support, love and encouragement.

Last but not least, I would like to give a special mention to my lovely wife, Hanh. She has, without a doubt, been amazingly supportive not just during my PhD, but throughout my life. Additionally, I feel so thankful to have my two little angels, Mymy and Mia, who has given such an incredible moral support for me to accomplish my PhD journey.

Chapter 1

Introduction

Countries around the world currently face two main greatest challenges: reducing carbon emissions and securing sustainable energy supplies (US Department of Energy, 2003; Department of Energy & Climate Change, 2009a). With current trends relating to the increase of energy use year on year, the world demand is predicted to increase by more than 90% by 2035, and the share of fossil fuels in the world's energy mix falls from 82% to 76% in 2035 (International Energy Agency, 2013). It is therefore necessary to investigate methods to improve the energy infrastructure to meet this growing demand without causing irreversible damage to the environment.

In an effort to mitigate the adverse effects of climate change, many countries have committed to transition to a low carbon economy. Specifically, the UK has legislated that carbon emissions should be reduced by 80% by 2050 (compared to their 1990 baseline) (Department of Energy & Climate Change, 2009b). To meet this target, a key aim is to improve energy efficiency through the use of two main sources: heating living spaces and (about 13% of all UK CO_2 emissions) and transportation sector (about 22% of all UK CO_2 emissions) (Department of Energy & Climate Change, 2009b). In particular, more efficient heating technologies such as air or ground source heat pumps powered by electricity will be used rather than using natural gas or oil (MacKay, 2007). Within the transport section, ultra-low carbon vehicles will become widespread. As a result, the demand for electricity will be increased dramatically. To meet this electricity demand, there is a need to increase low-carbon electricity generators which use renewable resources such as wind and solar into electricity networks.

Meeting this increased demand for electricity presents a major challenge to existing electricity grids that need to maintain the perfect balance between supply and demand. More specifically, the electricity distributed from a relatively small number of large fossil fuel burning power stations to millions of consumers, and the central operation on this original electricity grid is based on the idea that supply must always follows demand. This balance is very crucial in operating an electricity system. To guarantee the balance between the demand and supply, the supply-side must be varied in real-time (by adjusting the output of generators such as voltage and frequency across the grid) to match the demand-side requirements. Maintaining this regime where supply follows demand has several negative implications. First, the generation of electricity is likely to be inefficient. In particular, to satisfy peak demand at specific times of the day on a daily basis, a number of additional generators are required to generate power at short notice (i.e., spinning reserve). This requirement of additional generators results in the real cost of generating electricity to vary widely throughout the day, while, on the other hand, the electricity price charged to consumers is typically much cheaper. As a result, the whole operational process is inefficient. Second, there will be significant losses within the transmission and distribution networks due to ageing infrastructure (e.g., 40-year old power stations). Third, with the increasing penetration of renewable energy generation embedded in the distribution network (e.g., community wind turbines, or solar panels on houses), the generation of electricity may become more unpredictable and unmanageable. As a consequence, flows on the distribution network will also be more uncertain. To address such issues, investment costs to reinforce the current grid infrastructure can be huge and infeasible. Instead, fundamental operation of the grid might need to be redesigned to increase efficiency between supply and demand within the grid.

To increase electricity efficiency within the grid, the demand-side will need to be managed to adapt to the available supply. This control regime is termed 'demand follows supply'. Indeed, the idea of controlling loads and generators to respond to network constraints and price signals have been highlighted for a number of reasons (Schweppe et al., 1989). First, peaks in demand can be 'flattened' and this, in turn, reduces the need to operate costly and carbon intensive peaking power plants. In the long term, this flattening allows cheaper cost to the consumers, and more economically viable for businesses that require greater supply reliability and flexibility. Second, the grid operators can be flexible and quickly recover if there are any unforeseen events (e.g., the unavailability of renewable energy sources).

This drive for demand-side management (DSM) is one of the key elements of the vision of 'smart grid', whereby energy demand can be controlled using incentives and automation. Research has shown that even small shifts in peak demand can have significant impacts on savings for consumers (Spees and Lave, 2008). In more detail, the smart grid is described by the US Department of Energy (US Department of Energy, 2003) as "a fully automated power delivery network that monitors and controls every customer and node, ensuring a two-way flow of electricity and information between the power plant and the appliance, and all points in between. Its distributed intelligence, coupled with broadband communications and automated control systems, enables real-time market transactions and seamless interfaces among people, buildings, industrial plants, generation facilities, and the electric network". Consequently, smart grids intelligently integrate the actions of all participants in the system to deliver reliable, low carbon electricity. Moreover, smart grids use smart metering technology, which is set to open up a range of new services to consumers to allow greater choices and control over their energy usage. These meters will also provide more accurate bills, support faster and easier supplier switching, and allow suppliers to offer a greater range of tariff packages and services, as well as facilitating the control of appliances within the home. In particular, in the *domestic sector*, the UK Government has already committed to ensuring that smart meters for both electricity and gas are in all homes (47 millions meters in 26 million properties) by 2020 (Department of Energy & Climate Change, 2009a). More specifically, an investment of 8.6 billion pounds will be spent in replacing some 47 million gas and electricity meters, which are expected to deliver total benefits of 14.6 billion over the next 20 years (compared to 2009 level) (Department of Energy & Climate Change, 2009a). Similar commitment has also been shown in the US in which twenty-eight US utilities are committed to rolling out smart meters to their customers in the next few years (Department of Energy & Climate Change, 2009a).

At the domestic level, figures show that domestic electricity consumption accounts for approximately 27% of worldwide electricity consumption (International Energy Agency, 2008). Moreover in the UK in 2009, domestic electricity usage accounted for approximately 24% of the country's overall electricity consumption (Department of Energy & Climate Change, 2010). Particularly, deferrable loads[1] already accounted for approximately 20% of the domestic electricity usage, and is likely to grow due to the increased electrification of space and water heating (MacKay, 2007; Department of Energy & Climate Change, 2009b). Hence, it is expected that DSM will have a major impact on CO_2 emissions and overall generation costs, as minimising peaks in consumption reduces the need for expensive peaking plants that can be also be highly carbon intensive. More importantly, DSM is enhanced by the concept of the future smart home in which advanced automation systems and appliances such as lighting, heating, key electronic devices and services can be remotely controlled, monitored and accessed to manage electricity use in order to maximise users' comfort and savings while avoiding peaks on the grid. For example, a smart home can control lighting, temperature, multi-media, turn on central heating at the economical periods that are most comfortable for the home's users. We discuss DSM methods to manage electricity demand at a large scale in the following section.

1.1 Agent-based Demand Side Management

The scenario we study in this thesis involves some of the operations of a future 'smart' home (see Figure 1.1). In particular, within this home, electricity is supplied by both distant generators (e.g., delivering electricity over transmission and distribution networks),

[1] The use of the appliances that can be delayed/shifted.

and local energy sources through domestic micro-generation based on low-carbon resources such as solar panels or wind turbines, and micro-storage (such as a battery) where electricity is stored and can be re-used to maximise the electricity usage from intermittent sources (e.g., wind or solar).

In addition, individual home devices will be monitored by sensors allowing for intelligent home automation (e.g., lights can be turned off if no one is at home, or the electric heaters could be scheduled to turn off when the price of electricity is high). Moreover, home automation systems (e.g., Samsung Smart Home Appliances,[2] Microsoft Hohm,[3] Intel Home Dashboard Concept[4]) will provide sensors to control home appliances and monitor local weather conditions in order to optimise heating and electricity usage.

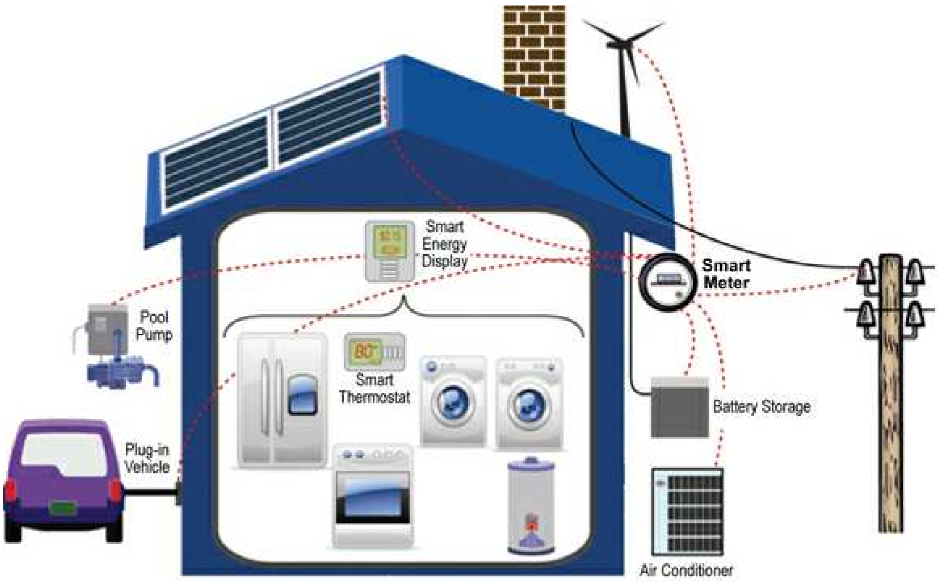

Figure 1.1: An example of smart home model. (source: http://www.digikey.com)

Now, the advent of the smart grid with two way information flows will potentially allow smart homes to automatically manage loads *via* the smart meter in reaction to grid conditions.[5] In particular, based on the real–time carbon intensity (see Figure 1.2) or the real–time price of electricity, grid operators may send signals to the smart meter in the smart home in order to curtail demand or coordinate the flow of electricity (Ramchurn et al., 2011a). Theoretically, if the price of electricity (or CO_2) is high on the wholesale energy market, the utilities would motivate smart homes to reduce or shift their consumption loads. In more detail, consumers may be charged more for using electricity at peak hours using tariffs such as time-of-use (TOU) (e.g., Economy 7, Economy 10 in the UK), or real-time pricing (RTP) (e.g., the GridWise Project[6]). Home devices may

[2] http://www.samsung.com/uk/home-appliances/.
[3] www.microsoft-hohm.com.
[4] http://www.intel.com/p/en_US/embedded/applications/energy/energy-management.
[5] We describe the smart meter in more detail in Section 2.1.1.
[6] See more details at http://gridwise.pnl.gov/.

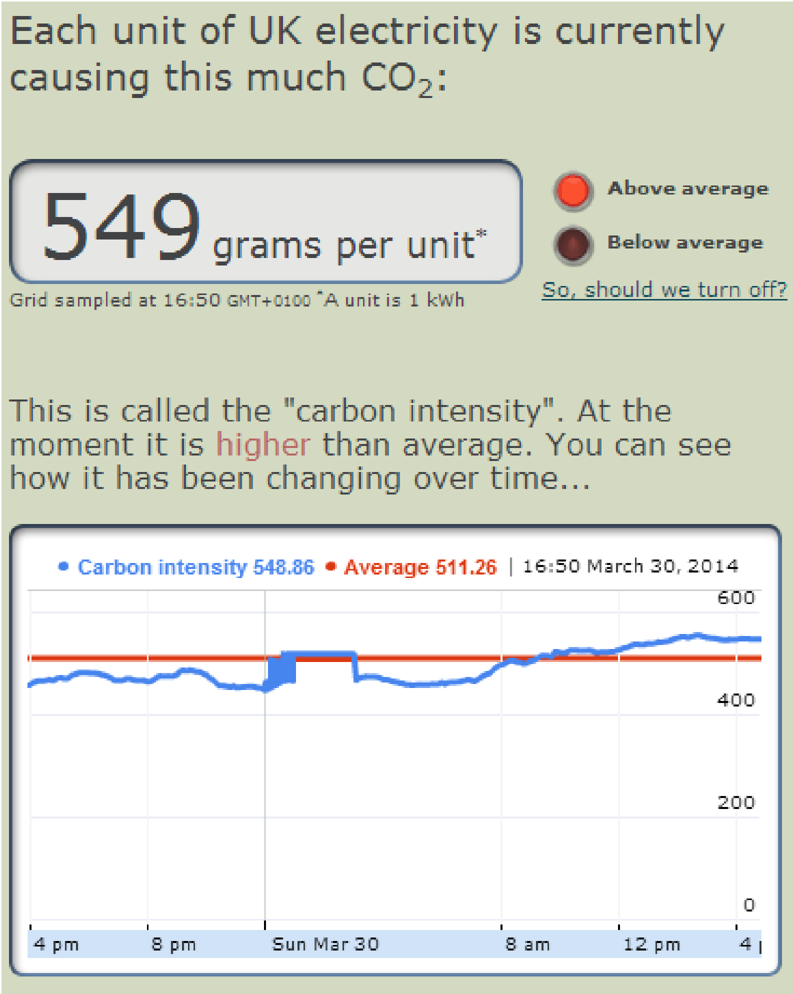

Figure 1.2: An example of a display of real-time carbon intensity in the UK (source: http://realtimecarbon.org/).

also be directly controlled using signals from the grid (e.g., the frequency of AC power). For example, air conditioners may be automatically turned off at peak time, or storage heaters may be turned on at night using radio signals (Faruqui and George, 2005). Such techniques assume that providing more automation and incentives to consumers helps them manage the use of appliances more efficiently. However, this assumption is challenged by the two main problems: i) the signals from the grid are typically too complex for the average user to take any meaningful action as a result (Schweppe et al., 1989); ii) consumers need to perform complex calculations in order to obtain an optimal plan (e.g., schedule their appliance runs or activities according to a real-time price profile) that maximises their savings (Costanza et al., 2014).

Against this background, a set of agent–based demand–side management (ADSM) techniques have been proposed to address these challenges (Vytelingum et al., 2011; Ygge and Akkermans, 1999; Ramchurn et al., 2011a; Kashif et al., 2011). In particular, these methods involve deploying autonomous–software agents in smart meters in the home, acting on behalf of the user. By taking into account the real time carbon content/cost of electricity, the agent can schedule appliance usage on behalf of the user in order to minimise peak demand. However, in so doing, it is also essential to take into account the user's preferences as well as the user's typical consumption patterns (e.g., the use of washing machine implies the use of the clothes dryer) (or the inter–dependencies between appliance usage) in the home when scheduling loads (Rodden et al., 2013; Alan et al., 2014; Fischer et al., 2013). However, these recent ADSM techniques simplify the user's preferences in the optimisation of the scheduling plan to an arbitrarily weighted function of cost and comfort variables. Thus, such scheduling methods may not be acceptable to users as the plans are not compatible with their everyday routine. For example, suppose that a user prefers to use the washing machine on weekends when she/he has time to take clothes out to dry and iron them. As a consequence, she/he would not accept a suggestion to use the washing machine on a week day, even though it may be cheaper to do so. Moreover, demand–side management algorithms generally ignore inter–dependencies between the usage of different appliances (see Section 2.1.2 for more details). For instance, a user might use the dishwasher and the oven on the same day, or prefers to turn on the TV whenever she/he starts cooking. To produce rescheduling suggestions that meet a user's preferences (and therefore more likely to be acceptable), it is crucial to *forecast* their appliance usage by capturing both the user's habits (or behaviour) and the inter–dependencies between appliances. Taking such forecasts into account, an agent would be able to empower the users to make informed energy reduction decisions by providing more *personalised* recommendations on how to use appliances to reduce cost and CO_2 emissions. We discuss the appliance usage prediction problem in the next section.

1.2 The Appliance Usage Prediction Problem

This section describes the challenges of the appliance usage prediction problem in the smart home, which is considered as a key building block to improve the efficiency of agent-based DSM techniques in real–world deployments. By solving the appliance prediction problem, the agent can forecast which appliances are likely to be used, and when they are likely to be used at a certain point during a day in the future. Taking such forecasts into account, the agent can optimise the future plan (e.g., a day ahead, or a week ahead) for appliance usage based on minimising electricity costs (or CO_2) and the user's preferences. In addition, the agent can also provide *personally meaningful* suggestions to the user to maximise their savings (i.e., money). More importantly, in order to forecast appliance usage accurately, appliance usage prediction algorithms need to take into account both the human routine activities (e.g., a particular user might prefer using their washing machine on weekend and watching TV in the evening during weekdays) and the inter–dependencies between appliances (e.g., the use of washing machine implies the use of dryers, and the use of ovens and microwaves implies the use of dish washers).

To date, modelling inter-dependencies between appliances has been approached using mainly two different techniques: i) a rule-based method based on Frequent Episodes Discovery; ii) Temporal Point Processes which is a stochastic process composed of a time-series of binary events (Daley and Vere-Jones, 2003).[7] However, as we will show later in Chapter 6, by not considering human routines, these approaches do not perform well on real-world datasets of energy consumption from homes.

In contrast, methods for human behaviour prediction have been developed for many applications such as activity recognition, location prediction and human presence prediction (González et al., 2008; McInerney et al., 2013; Tominaga et al., 2012). These algorithms consider the pattern of human routines but have not been specifically modelled complex inter-dependencies between the usage of different appliances within a typical home. For example, location prediction has to deal with only one data stream, while a key difference is that in our domain of application there are multiple concurrent data streams, one per appliance.[8]

Furthermore, to deploy the appliance usage prediction algorithms in a real–world application, the prediction accuracy is one of the most important factors to help users to trust the system. Without this trust, users would ignore intelligent features such as personalised recommendations from the system. One example of such loss of trust is the deployment of the Nest thermostat (Yang and Newman, 2013), which features an attractive wall-mounted device as well as smart phone and web-based control capabilities. The Nest device utilises such machine learning and sensing technology to learn users' comfort and maximise energy savings as well as convenience and enjoyable interaction. Field trial studies of the Nest thermostat in real houses show that users are not satisfied with the device mainly due to the inability to accurately learn and predict users' comfort and scheduling of the thermostat usage.

Against this background, the goal of this thesis is to address the aforementioned challenges of the appliance usage prediction problem. Moreover, we also develop an Intelligent Demand Response mechanism (see Chapter 7); a simulated scenario that makes use of the appliance usage prediction algorithms we develop, to demonstrate the process of maximising monetary savings and comfort for the user (or home's users). In the next section, we provide our research requirements for this prediction problem.

1.3 Research Requirements

An algorithm for appliance usage prediction has to satisfy the following broad requirements:

[7] Related work for modelling inter-dependencies algorithms will be described in Chapter 2.2.

[8] We discuss related work for prediction human routine in Section 2.3.

1. **Human routine behaviour**: it is important to be able to model user routines, taking into account the use of appliances on specific days of the week or times of the day, in order to make more accurate predictions.

2. **Ability to learn inter–dependencies between appliances**: as discussed above, consumers may use multiple appliances in temporal sequences that reflect interdependencies between the appliance usage activities. Hence, our algorithms need to be able to model such temporal inter–dependencies accurately.

3. **Prediction accuracy**: given models of appliance usage and human routines, it is important to develop methods to learn to predict the usage of appliances with high accuracy in order to avoid annoying the user with pointless suggestions to shift or reduce appliance usage in the future.

4. **Computational efficiency**: algorithms for appliance usage prediction should be able to return predictions at short notice in the case where users need to be notified to shift or reduce consumption in real-time.

In the next section, we discuss our contributions against these requirements.

1.4 Research Contributions

Given the requirements described in Section 1.3, our research aim is to develop the algorithms to tackle the appliance usage prediction problem. In so doing, we contribute to the state–of–the–art and meet all four requirements by developing a number of novel appliance usage prediction algorithms. More specifically, we propose novel algorithms to predict the usage of appliances that take into account both daily practices of users and the inter–dependency between their usage of different appliances. Our algorithms take advantage of the cyclic nature of routine activities (González et al., 2008). These contributions are summarised in Table 1.1. Moreover, we identify a trade–off between prediction accuracy and computational cost, and hence propose two algorithms that satisfy the needs for fast prediction and high accuracy respectively. In more detail, to generate predictions of appliance usage in limited time, we develop *EGH-H*, a rule–based method to find significant patterns in the energy consumption activities that can capture the user's behaviour. In particular, we build a model that models the cyclic behaviour of the users and hence use the Episode Generation Hidden Markov Model (EGH) (Laxman et al., 2008) to efficiently capture the patterns that form the inter–dependency between the usage of the appliances. To provide more accurate predictions, we propose a graphical–model based approach, *GM-PMA*. In particular, we use the Dirichlet Process Mixture (DPM) model to identify a set of latent classes that represent sets of behaviour of the user (e.g., working days, holidays, weekends), conditioning on the weekly periodicities of routine behaviour. These latent classes are never directly observed, and can capture the inter–dependencies between appliance usage at different times of the day. We empirically evaluate EGH and GM-PMA and benchmark them against other comparable algorithms on two real–world datasets: the FigureEnergy dataset (Costanza et al., 2012), and the REDD dataset (Kolter and Johnson, 2011), and show that our approach outperforms them significantly.

Furthermore, we study a particular application of our algorithm to improve demand–side management. In more detail, we simulate a scenario within the home in which an agent can assist the user to optimise their savings and comfort using our appliance usage prediction algorithm. By taking into account real–time energy costs, the agent can suggest the deferment of some energy consumption activities to minimise overall cost while minimising discomfort. Crucially, the agent can learn to provide more meaningful suggestions by learning from the reaction of the user to previous deferment suggestions. Thus, this thesis advances the state–of–the–art as follows:

- We propose GM-PMA, the first, graphical model based, algorithm that can model both human behaviour and appliance usage inter–dependencies to efficiently predict the usage of electrical appliances in the home. Moreover, considering the trade-off between computational cost and accuracy of prediction, we also propose EGH-H, a faster, rule–based, algorithm that can integrate cyclic human behaviours and capture inter–dependency of patterns over activity streams.
- We demonstrate through extensive empirical evaluation, using real–world data from the FigureEnergy dataset and the REDD dataset, that GM-PMA outperforms existing algorithms by up to 41% in terms of prediction accuracy. We also show that the runtime of EGH-H is typically 100 times faster than that of the benchmarks.
- We propose an Intelligent Demand Response (IDR) mechanism that an agent uses to assist the user to improve their savings and comfort, by using an appliance usage prediction algorithm. In addition, we simulate different user behaviours and demonstrate that the agent can learn a user's preferences from their responses to the agent's suggestions. We empirically evaluate the performance of IDR with a number of different simulated user behaviour, and show that, by using IDR, users are likely to improve their savings significantly.

The work up until now has led to following contributions to the literature:

- Truong, Ngoc Cuong, McInerney, James, Tran-Thanh, Long, Costanza, Enrico and Ramchurn, Sarvapali D. (2013) Forecasting Multi-Appliance Usage for Smart Home Energy Management. In, *Proceeding IJCAI'13 Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2908-2914, Beijing, China.

	EGH-H	GM-PMA
Human routine behaviour	++	++(*)
Ability to learn inter–dependencies between appliances	++	++(*)
Prediction accuracy	+	++(*)
Computational efficiency	++(*)	+

Table 1.1: An overview of our contributions in terms of the research requirements in the appliance usage prediction problem. The symbols have the following meaning: '+' ('++') means that the requirement is (strongly) satisfied. In addition, '(*)' indicates the best performance of the row. On the other hand, '-' means the requirement is not satisfied.

- Truong, Ngoc Cuong, Tran-Thanh, Long, Costanza, Enrico and Ramchurn, Sarvapali D. (2013) Towards Appliance Usage Prediction for Home Energy Management. In *Proceedings of the fourth international conference on Future energy systems*, pages 287-288, ACM, 2013.
- Truong, Ngoc Cuong, Tran-Thanh, Long, Costanza, Enrico and Ramchurn, D. Sarvapali (2013) Activity Prediction for Agent-based Home Energy Management. In, *Agent Technologies for Energy Systems (ATES 2013).*
- Truong, Ngoc Cuong, Tran-Thanh, Long, Costanza, Enrico and Ramchurn, Sarvapali D. (2012) Predicting energy consumption activities for home energy management. At *Agent Technologies for Energy Systems (ATES 2012)*, Valencia, ES.
- Truong, Ngoc Cuong, McInerney, James, Tran-Thanh, Long, Costanza, Enrico and Ramchurn, Sarvapali D. (2014) Algorithms for Appliance Usage Prediction. In, *Journal of Artificial Intelligence Research (JAIR) 2014* (under review).

1.5 Report Outline

The remainder of this thesis is organised as follows:

- Chapter 2 discusses the vision of the smart grid, and the problem of managing demand in the future smart home. First, we discuss the agent-based demand-side management as a key technique to improve efficiency significantly in DSM as well as managing even thousands of smart homes at large scale. Second, we provides related work for modelling and predicting inter-dependencies between appliances (or events) in temporal data. In particular, we describe two main approaches that can be used to model the inter-dependencies between appliances: a temporal point process technique, and a rule-based method based on Frequent Episode Discovery. These algorithms will be used as benchmarks later in Chapter 6. Then, we discuss existing work on predicting human routines in general. Lastly, we provide general knowledge and techniques for probabilistic graphical models that we will use throughout this thesis.

- Chapter 3 describes our appliance usage prediction models, and the real–world dataset that we use for evaluation. First, we mathematically model the appliance usage problem. Second, we describe two real–world datasets, and provide methods for data preparation before evaluation.
- Chapter 4 describes our rule-based algorithm, EGH-H, that is applicable for scenarios in which notifications need to be generated in a short time. First, we describe the inter-dependency model between appliance usage. Second, we provide the human routine model that takes into account human routine behaviours. Third, we describe the mixture model to improve the prediction performance. Last, we provide the appliance usage prediction to forecast the future appliance usage.
- Chapter 5 describes our graphical model based algorithm, GM-PMA, that meets research requirements 1 to 3 but trade–off computational efficiency in order to provide more accurate predictions of appliance usage than EGH-H. First, we provide the clustering inter-dependency model to identify the behaviour of any given observation. Second, we describe how to use Gibbs sampling inference to estimate the parameters of the model. Last, the appliance usage prediction is provided based on marginalisation from unknown variables.
- Chapter 6 shows empirical evaluations for our algorithms, comparing other benchmarks. First, we give the settings set up for the experiments. Second, we describe all the benchmark algorithms. Third, we provide the evaluation metrics to measure the performance of the algorithms. Fourth, we evaluate our algorithms with a synthetic dataset. Last, we empirically evaluate the prediction accuracy and the runtime of all algorithms on the real-world datasets.
- Chapter 7 proposes an Intelligent Demand Response (IDR), a simulated scenario, to make use of our appliance usage prediction algorithms. First, we provide the IDR's mechanism in more detail. Second, we describe how to simulate human responses to suggestions. Third, we discuss the agent to optimise personalised suggestions to the users. Fourth, we describe logistic inference which is used to learn user's preferences based on user's responses. Last, we empirically evaluate the IDR mechanism using the real-world dataset.
- Chapter 8 summarises the key outcomes of this thesis and the conclusions that can be drawn from each chapter. We also discuss theoretical and practical implications of our work.

Chapter 2

Literature Review

In this chapter, we provide an accounted background research related to this thesis, and the necessary theory in developing the algorithms presented in Chapters 3, 4 and 5. Section 2.1 describes a vision of the smart grid with a focus on demand–side management of domestic consumption, discusses the demand–side management challenges that might raise in future smart homes and reviews related work that has been proposed to address some of these challenges. Section 2.2 discusses existing algorithms for prediction from inter–dependent data streams, by elaborating on two main approaches for modelling and predicting the inter–dependencies between activities: i) Temporal Point Processes, and ii) a rule-based method based on Frequent Episode Discovery framework. Section 2.3 provides related techniques for modelling and predicting human routine behaviour. Section 2.4 provides background of probabilistic graphical models, describes Hidden Markov Models and some approximate inference techniques that are used for discussion in developing our algorithms in Chapter 4 and 5. Finally, Section 2.5 summarises the key concepts described in this chapter and justifies the techniques that are built upon within this thesis.

2.1 The Smart Grid

As discussed in Chapter 1, many countries have committed to transition to a low carbon economy. In particular, the UK has legislated to achieve an 80% reduction of CO_2 emissions by 2050 (Department of Energy & Climate Change, 2009a). To achieve such ambitious targets, current electricity grids will have to be modernised in order to cope with increasing demand and intermittent generation from renewable energy sources.[1] However, a wholesale hardware upgrade would just be too costly. Given this, the smart

[1]Detailed motivation has been discussed in Chapter 1.

grid has been proposed to offer a prospect of cost-effective and more efficient management of the energy infrastructure. In particular, the US Department of Energy (2003) describes a smart grid as:

> A fully automated power delivery network that monitors and controls every customer and node, ensuring a two-way flow of electricity and information between the power plant and the appliance, and all points in between. Its distributed intelligence, coupled with broadband communications and automated control systems, enables real-time market transactions and seamless interfaces among people, buildings, industrial plants, generation facilities, and the electric network.

Note, here, the particular focus on automation and interfaces that will create opportunities for a new range of services to intelligently coordinate the actions of all participants (e.g., consumers, devices, suppliers) in the system and, by so doing, specifically allow for autonomous demand response (i.e., the management of demand in relation to the prevailing supply conditions). Furthermore, with two-way flows of real–time information on the grid network, the smart grid promotes interaction between suppliers and consumers. The suppliers can: i) provide *dynamic* tariffs that reflect the costs (and carbon content) of generation; ii) or reward consumers for using more electricity at times when there is a lot of electricity output generated from renewable sources. This transparent information also allows the electricity network operators to improve the management of electricity use by having a better coordination between supply and demand.

In more detail, the Department of Energy & Climate Change (2009a) defines four key principles underlying such a smart grid as follows:

- **Observable**: provides a wide range of real–time operational statistics such as the condition of equipment and technical information in both the transmission network and distribution network.
- **Controllable**: provide better management and optimisation between demand and supply on a large scale as well as incorporating the use of intermittent renewable generation sources.
- **Automated**: include a number of autonomously intelligent features that will be able to make certain automatic demand response decisions. It may also be able to reconfigure its settings in response to power fluctuations or outages.
- **Fully integrated**: fully adapted to existing electricity systems and compatible with other new devices such as smart home appliances.

Now, the intelligence implied by such definitions of the smart grid tend to naturally relate to the concept of the smart home, in which key appliances and services can be

Figure 2.1: A ‘Liberty’ smart meter manufactured by Secure Meters UK Ltd.

remotely monitored and controlled through the smart meter in order to optimise the management of electricity use to maximise the user’s comfort and savings in the home while avoiding peaks on the grid. Hence, we next discuss the use of the smart meters in more detail in Section 2.1.1. The key principles of the smart grid will underpin demand–side management (DSM) in which electricity use can be monitored and adapted in order to minimise peaks on the grid by either remotely controlling appliances or incentivising customers to shift/reduce their consumption at certain times. We discuss DSM in more detail in Section 2.1.2.

2.1.1 Smart Meters

Central to the vision of the smart grid is the deployment of smart meters that promise to allow greater control over electricity use for consumers. A smart meter (see Figure 2.1) is an electronic device that records power or energy consumption in intervals of an hour or less and communicate via a network back to the utility for monitoring and billing purposes (The Climate Group, 2008). In the UK, smart meters typically collect energy consumption data at half-hourly intervals and transmit this data to the utility (Smart Metering Design Group, 2011). Some countries have already committed to roll out smart meters to consumers’ homes. For example, the UK Government has committed to ensuring that 47M smart meters will be installed in 26M homes by 2020 (Department of Energy & Climate Change, 2009a). In the US, twenty-eight US utilities have committed to rolling out smart meters to their customers in the next few years (Department of Energy & Climate Change, 2009a).

Figure 2.2: The AlertMe's meter reading kits.

In more detail, smart meters consist of two parts: a metering module, and a communication module. The metering module involves measuring, monitoring, and storing energy (or electricity) consumption data at specified intervals. Apart from meter reading products that are installed by the utility operators, there are several existing off-the-shelf products available for the in-home monitoring such as Plogg,[2] AlertMe,[3] Plugwise.[4] These off-the-shelf meter reading products usually offer ZigBee (IEEE 802.15.4) wireless-mesh-based smart-plug units used in-line with appliances that are designed specifically for in-home energy monitoring. For example, AlertMe's meter reading kits (see Figure 2.2) measure power consumption of the whole house by clipping a clamp meter through the standard mains voltage (or measure consumption of the appliance using a AlertMe Smart Plug device). The clamp and the Smart Plug are connected to a battery-powered wireless transmitter. Then, consumption data from the meter clamp and the Smart Plug are transmitted to a supplied hub (which plugs into an ADSL or cable routine) and the internet. In turn, the communication module offers two-way communications between suppliers and consumers. The communication module communicates to the utility operators through GPRS technology and in the case of off-the-shelf products, this module typically uses a broadband connection.[5] Information such as real–time electricity bills, electricity prices, CO_2 emissions can be provided from the utility operators to the smart meter.

Smart meters usually include an *in-home display* (IHD), in which signals are sent to a display panel from a transponder attached to the meter tail, to allow consumers to monitor their energy consumption in real-time and retrospectively. Darby (2006) shows that in-home displays help interested users better understand and manage their electricity use through feedback information (on real-time and historic usage), and hence achieving savings in the range of 5–15%. Recent real–world applications in relation to in-home display have been deployed to help consumers manage their electricity use such

[2]http://www.plogg.co.uk.
[3]https://www.alertme.com.
[4]http://www.plugwise.com/.
[5]http://www.securetogether.com/.

as Navetas,[6] and AlertMe.[7] Also, the IHD, allied with smart meters, can be developed to encourage behaviour change by showing consumers how their energy consumption patterns compared with those in similar households, then offers personalised tips to save more energy (Froehlich et al., 2010).[8]

Moreover, a key capability of smart meters is that they allow suppliers to implement demand–side management programmes. In more detail, energy supply companies may offer their customers varying tariffs through the day that reflect the amount of supply available on the system via the smart meter to incentivise them to switch some non-urgent energy consumption activities away from peak times. By so doing, consumers can save money on their bills, and CO_2 emissions can also be minimised (by minimising the peak demand). In addition, combined with the appliances in the home that can be remotely controlled via smart meters, electricity use can be managed to maximise consumers' savings and comfort while minimising peaks on the grid. We discuss the aspect of demand–side management in more detail in the next section.

2.1.2 Demand-Side Management

DSM involves the planning, implementation and monitoring of electricity consumption activities in order to alter the use of their electricity in ways that will produce desired changes in demand (Gellings, 1985; Strbac, 2008). Typical DSM approaches involve a central controller that incentivises consumers to reduce their demand at peak times. These approaches have been limited to two techniques, namely direct load control (DLC), and smart pricing mechanism (SPM). DLC involves remotely controlling home appliances using signals from the grid (e.g., the frequency of AC power) (Ruiz et al., 2009; Gomes et al., 2007; Weers and Shamsedin, 1987; Chu et al., 2005). For example, air conditioners may be automatically turned off at peak times, or storage heaters may be turned on at night using radio signals (Faruqui and George, 2005). However, it has been shown that the users' privacy can be a major barrier in implementing DLC (OpenHAN Task Force of the Utility AMI Working Group, 2010). SPM, in turn, involves consumers being charged more for using electricity at peak hours. In this regard, time-of-use (TOU), critical-peak pricing (CPP), and real-time pricing (RTP) are popular pricing mechanisms. However, only deploying these mechanisms to consumers is ineffective because additional peaks in demand can be created instead. For example, in TOU implementation (e.g., Economy 7, Economy 10, or eco:2020 in the UK); where the electricity price during the day is more expensive than the electricity price at night, it has been observed that significant additional peaks in demands are created at off-peak hours (Strbac, 2008). In turn, CPP is often applied to control air-conditioners at

[6] http://www.navetas.com/.
[7] https://www.alertme.com/.
[8] This idea is also known as peer-pressure, which has been deployed recently. More detail can be seen at http://opower.com/solutions/demand-response.

peak periods in the USA, can raise additional peaks as soon as the critical period is over. Similar results are caused by using RTP pricing scheme (e.g., the demonstration of Grid-Wise Project[9]) in which the price of electricity varies at different hours (or half-hours) of the day. Moreover, these approaches assume that the appliance use may be improved efficiently if consumers are provided more automation and incentives. However, this assumption raises two challenges: i) the average user can not take any meaningful action as the signals from the grid are typically too complex (Schweppe et al., 1989); ii) in order to maximise savings, consumers may need to perform complex calculations (Costanza et al., 2014). Also, these techniques will not be scalable to deal with millions of homes or buildings national-wide (Ramchurn et al., 2011a).

To address these challenges, recently, a number of agent-based demand-side management techniques (ADSM) have been proposed (Deindl et al., 2008; Ramchurn et al., 2011b,a; Vytelingum et al., 2010; Ygge et al., 1999; Wong et al., 2010). These approaches involve deploying autonomous–software agents in smart meters in the home, acting on behalf of the user, assist the user to manage the appliance usage more efficiently. By taking into account the real time carbon content (or cost of electricity), an agent can schedule appliance usage in order to avoid peak demands. In particular, Wong et al. (2010) use game theory to formulate an energy consumption scheduling game in which the players are the users and their strategies are the daily schedules of their household appliances and loads. Vytelingum et al. (2010) propose the ADSM technique by presenting a model of smart meters controlling the use of energy storage devices in homes. In more detail, they optimise the usage of users' activities and the storage profile of the house using various information sources (e.g., predicted energy via weather data, supplier's price plan). By so doing, they provide agent-based micro-storage strategies to learn the best storage profile to adopt so that the agents could learn to buy most profitable storage capacity while helping to minimise peaks on the grid. In terms of deferrable load management, Ramchurn et al. (2011a) propose a mechanism for agents to *adapt* (instead of reacting as in traditional DSM mechanisms) the deferment of their loads to optimise the use of electrical loads and minimise peaks. By using average UK consumption profiles for 26M homes, they show that the peak demand of domestic consumers in the grid can be reduced by up to 17% and carbon emissions by up to 6%.

These agent-based approaches, unfortunately, do not take into account the user's preferences as well as the user's typical consumption patterns (e.g., the use of washing machines implies the use of the tumble dryers) (or the inter–dependencies between appliance usage) in the home when scheduling loads (Rodden et al., 2013; Alan et al., 2014). Thus, such scheduling methods may not be acceptable to users as the plans are not compatible with their everyday routine (Rodden et al., 2013; Alan et al., 2014). For example, suppose that a user prefers to use the washing machine on weekends when she/he has time to take the clothes out to dry and iron them. Consequently, she/he

[9]See more details at http://gridwise.pnl.gov/.

would not accept a suggestion to use the washing machine on a week day, even though it may be cheaper to do so.

Moreover, demand–side management algorithms generally ignore inter–dependencies between the usage of different appliances. For instance, a user might use the dishwasher and the oven on the same day, or prefers to turn on the TV whenever she/he starts cooking. As a result, suggested schedules that do not take these inter–dependencies into account may not be adapted to the user's behaviours, and thus, not be accepted. To produce rescheduling suggestions that meet a user's preferences (and therefore more likely to be acceptable), it is crucial to forecast the appliance usage by capturing both the user's habits (or behaviour) and the inter–dependencies between appliances. Taking such forecasts into account, an agent would be able to empower the users to make informed energy reduction decisions by providing more personalised suggestions on how to use appliances to reduce cost and CO_2 emissions.

To address this limitation, in this thesis, we address the challenge of modelling appliance usage and predict future activities (see Chapter 4 and 5). In order to understand the key contributions of this thesis, we next study related works in terms of predicting dependencies in temporal data in Section 2.2, and predicting human routines in Section 2.3.

2.2 Prediction with Inter–Dependent Data

In this section, we review two main popular approaches that can be applied to model the inter–dependencies in time-series data (or event sequences). First, we discuss the use of temporal point processes in Section 2.2.1. Second, we discuss a data mining technique, specifically a rule-based method based on Frequent Episode Discovery to find the dependencies between variables (or appliances) in event sequences in Section 2.2.2. In addition, we use these methods as our benchmarks for our solution to the appliance usage prediction problem in Chapter 6.

2.2.1 Temporal Point Processes

Temporal point processes are widely used in many real–world applications to model the waiting time between two events based on the assumption that the present depends only on the past (Brown et al., 2004; Barbieri et al., 2001). In particular, a temporal point process is a stochastic (or random) process which is composed of a time-series of binary events that occur in continuous time (Daley and Vere-Jones, 2003). In addition, the whole point process can be described by the random interval between the points. In the domain of domestic sector, the duration of waiting time between two consecutive appliance usages is not constant over time. For example, a washing machine is frequently used over the weekend, and is rarely used on Monday. Instead, it could fluctuate depending on the usage of other appliances (or itself) in the past. Therefore, if we can model the waiting time between the use of appliances, appliance usage modelling can be done accurately, whereby appliance usage can be considered a temporal point process.

One such temporal point process model is the Poisson process (PP), which is a counting process characterised by a rate function $\lambda(t)$ that can be used to model the waiting time between the two consecutive events (i.e., appliance usage). The PP is called *homogeneous* if the rate function is constant over time. In this case, the waiting time between two consecutive events is exponentially distributed with rate λ as:

$$\forall t \in \mathbb{R}^+ : p(t|\lambda) = \lambda \exp(-\lambda t)$$

On the other hand, if the rate function is not constant over time, the PP is called *inhomogeneous* (Papoulis, 1991). In this case, the waiting time distribution is a generalised exponential distribution. In general, a PP can model the waiting time between the use of single appliance. However, it does not consider the inter-dependencies between the use of multiple appliances (Requirement 2).

To model the dependencies between different event types, Poisson Network Model (PNM) (Rajaram et al., 2005) and Piecewise-Constant Conditional Intensity Model (PCIM) (Gunawardana et al., 2011) has been proposed to learn the dependencies of each event type separately. Thus, in the appliance usage prediction problem, event types can be considered as different types of appliance such as dishwasher, tumble dryer, and washing machine. In more detail, PCIM captures the dependencies of each type of event based on historical events through a set of piecewise-constant conditional intensity functions, and represents these dependencies by decision trees. By capturing these dependencies, PCIM has been useful in forecasting the future interest of real web search users. However, these algorithms are not designed to exploit the behaviour of human users (Requirement 1), and thus, as we show later in Chapter 5, they fail in predicting human related data sequences (Requirement 3).

In the following section, we discuss an alternative approach to modelling dependencies in temporal event sequences.

2.2.2 Mining Dependencies in Temporal Event Sequences

In the appliance usage prediction problem, the dataset of the use of appliances in the home is a temporal sequence of events in which each event (i.e., appliance usage) has an associated time of occurrence. Within the dataset, there might be existing repeated occurrences of some patterns that provide insights into the user's everyday habits and the inter-dependencies between appliance usages (e.g., the pattern of the use of tumble dryers following by the use of washing machines). Given this, mining *meaningful* patterns in temporal event sequences can help to predict the next occurrences of appliance usage. In this context, Frequent Episodes Discovery (FED) is a popular framework for mining temporal patterns in event sequences (Mannila et al., 1997; Cho et al., 2010; Berberidis and Vlahavas, 2007; Tatavarty et al., 2007; Laxman et al., 2005). In this section, we review techniques for finding of frequent episodes in event sequences. First, Section 2.2.2.1 provides a brief overview of frequent episodes in event sequences. Second, Section 2.2.2.2 describes techniques to find frequent episodes. Section 2.2.2.3 discusses related work on using frequent episodes for prediction in time-series data. Lastly, Section 2.2.2.4 provides relevant background for the EGH algorithm.

2.2.2.1 Frequent Episodes in Event Sequences

This section briefly provides the framework of FED (Mannila et al., 1997). In general, the dataset of event sequences is written as $<(E_1, t_1), (E_2, t_2) \cdots >$, where E_i represents an *event type* and t_i is the time of occurrence of the i^{th} event. E_i takes a value from a finite set of event types. Please note that in the context of appliance usage prediction problem, the event is the appliance usage, and event types are the different types of appliances. For example, an event sequence containing 6 events is:

$$< (oven, 1), (cooking, 2), (dishwasher, 5), (oven, 7), (washingmachine, 9), (dryer, 12) >$$

An *episode* is an ordered tuple of event occurring together. For example, { *cooking* → *oven* → *dishwasher* } is a 3-node episode. An episode is said to occur in the event sequence if all events specified in the episode occur with the same time ordering in the event sequence. In addition, a *sub-episode* is a subsequence of the episode where the occurrences of events have the same ordering as the episode. *Frequency* of an episode is the fraction of windows in which the episode occurs. It can be defined in many different ways depending on the application scenario. A *frequent episode* is an episode in which its frequency exceeds a user-defined threshold.

Once frequent episodes are detected, they can be used to obtain rules that describe connections between events in the specific event sequence. We detail this problem in the next section.

2.2.2.2 Identify Frequent Episodes

The procedure of searching for frequent episodes applies the condition that, for an N–node episode to be frequent, all its (N-1)–node subepisodes should also be frequent (Srikant and R., 1995; Mannila et al., 1997). This search procedure is efficient level-wise, which starts from the episode with only one event in the first level, and then proceeding to the higher levels with increasing numbers of events. In particular, on each level, frequent episodes are processed in two phases: i) candidate generation, and ii) counting frequency.

In the candidate generation phase, a collection of candidate episodes is created by constructing every frequent episode, which is discovered from the previous level. The first level contains all episodes of size 1, which has only one activity.

In the counting frequency phase, each candidate episode is counted based on some predefined techniques (i.e., rules). In more detail, popular methods to find frequency of the episode include:

- **Window-based frequency**: is a fraction of number of occurrences of the episode in the given window over the length of the window. (Mannila et al., 1997)
- **Minimal occurrences of episodes frequency**: a minimal occurrence of an episode is defined as a window which there is no other occurrence of this episode that can be obtained in the sub-window. (Mannila et al., 1997)
- **Non-overlapped occurrence-based frequency**: *"two occurrences of an episode are said to be non-overlapping if no activity associated with one appears in between the activities associated with the other. The frequency of an episode is defined as the maximum number of non-overlapping occurrences of the episode in the activity sequence."* (Laxman et al., 2005)

Furthermore, the frequency of an episode is calculated based on counting the occurrences of the given episode (from the candidate episodes) within the event sequence using the counting rules (listed above). For example, for the episode *cooking* → *oven* → *dishwasher*, there would be an automaton that transits to state *cooking* when seeing an event type of *cooking* in the event sequence, and then waits for an event *oven* in the event sequence to transit to the next and so on. When this automaton transits to its final state, an occurrence of the episode is complete.

Having introduced an overall FED framework, now we turn to discuss this framework for prediction in temporal event sequences.

2.2.2.3 Frequent Episode Discovery for Prediction in Temporal Event Sequences

Predicting future occurrences of events in temporal event sequences can be achieved using a rule-based method that relies on FED to find *meaningful* patterns in temporal event sequences. In particular, these patterns can be used to explain the behaviour of the events in the past in order to generate the sequence rules. Then current events are matched to one of the generated rules to predict the occurrences of consequent events. In addition, finding *meaningful* episodes in temporal event sequences enhances a set of rules that can be representative of the occurrence of events and model the inter-dependencies between events.

In more detail, Weiss and Hirsh (1998) proposed a genetic algorithm to predict events by identifying the sequence of predictive patterns with constraints in historical data (e.g., predicting telecommunication equipment failures from network alarm data). These patterns represent inter–dependencies of events within time–series data. However, exploiting equally all data points of the historical data does not scale well. Given this, Laxman et al. (2008) show how to use rule–based methods to capture the relationship of the events within windows of events that precede occurrences of the target event types in historical data. In particular, they capture patterns (considering as frequent *episodes*) and exploit these *episodes* with the associated Hidden Markov Models (HMMs) to find the set of *significant* frequent episodes associated with each target event type. These sets of significant episodes are used to represent the behaviours in the historical data, and can be used to predict the future events. However, this method does not take into account the human behaviour (Requirement 1), and it fails to provide accuracy for appliance usage prediction (Requirement 3). We verify its performance in Chapter 6. Next, we provide relevant descriptions of the EGH algorithm.

2.2.2.4 The EGH algorithm

Considering an event stream $s =< E_1, E_2, \dots, E_n >$, where n is the current time instant, and E_i is an event type. Let Y denote the target event type that we want to predict its future occurrences. EGH will construct a prediction model to predict whether or not $E_{n+1} = Y$.

First, EGH constructs a training dataset for the target event type Y in the historical data s. A training dataset is constructed as follows. EGH examines all occurrences of Y to find the significant frequent episodes from the historical data s. Let K denote the number of occurrences of the Y in the historical data s. EGH builds a training dataset of event sequences $D_Y = \{X_1, \dots, X_K\}$, where X_i is the W-length slice of events from s, that immediately preceded the i^{th} occurrence of Y in s $(i = 1, \dots, K)$. Then, EGH identifies significant episodes in this training dataset (more details can be seen in Laxman et al. (2008)).

Second, EGH provides an associated specialised HMM to each significant episode. The HMM for the episode is constructed as follows. Let consider an N-node episode, $\alpha = (A_1, \dots, A_n)$. The associated HMM model contains 2N states. States 1 through N are the episode states, and states (N+1) to 2N are the noise states. The probability distribution for each episode state is the delta function $\delta_{A_i}(.)$. All transition probabilities are defined by a noise parameter $\eta \in (0, 1)$. In particular, the probability of transition into noise states is η, while the probability of transition into episode states is $(1 - \eta)$. In addition, the probability of the initial state is $(1 - \eta)$, and the probability of the last state (2N) is η. More details can be seen in Laxman et al. (2008).

Furthermore, the mixture model of all the significant episodes is constructed to predict the future occurrences of the given target event. We discuss this aspect in more detail in Chapter 4. Next, we discuss existing works on human routine prediction in the following section.

2.3 Predicting Human Routine

Human behaviour prediction algorithms have been developed for many applications such as activity recognition, location prediction (McInerney et al., 2013; González et al., 2008), mobility prediction (Eagle and Pentland, 2009) and daily routine prediction (Shimosaka et al., 2010; Tominaga et al., 2012). In the domain of mobility prediction, Eagle and Pentland (2009) use eigenvalue decomposition approach to extract the underlying structure in the daily patterns of human behaviour. They show that many people share a common set of mobility habits (e.g., going out of town over the weekend, leaving home to work in the morning, returning home in the evening). This common set of human habits is called a set of *eigenbehaviours*, and can be used to increase accuracy for new users. This approach is based on assumption that the future behaviour of the user follows the set of patterns that have been identified upon historical behaviour. In particular, the daily behaviour of a new user will be identified and approximated by a weighted sum of its eigenbehaviours. Hence, the prediction uses these weights that are calculated halfway through a day to predict the day's remaining behaviours. However, these approaches are designed to deal with sufficient, large-scale sets of training data. Also, it is arguable whether historical data can describe a full range of the user mobility patterns. Given this, McInerney et al. (2013) address the problem of predicting human behaviour with sparse data in the domain of location prediction. In particular, they developed a hierarchical Bayesian model based on shared probabilistic latent habits to exploit the similarities of new users with existing users. In turn, Farrahi and Gatica-Perez (2011) developed a methodology based on probabilistic topic models such as Latent Dirichlet Allocation topic model and the Author Topic model to detect the user's routines. Then, they use the discovered routines to determine behavioural patterns of users. However, these approaches are suited to transitioning between multiple mutually-exclusive labels (locations) rather than potentially concurrent activities. Hence, they do not satisfy our Requirement 2.

Moreover, Tominaga et al. (2012); Shimosaka et al. (2010) use a non-parametric method, Dirichlet Process Mixture (DPM) model, to extract the person's patterns of going out behaviour (i.e., to detect whether users are away from home). A number of daily patterns can be automatically discovered by the use of DPM model (as DPM model offers the flexibility for clustering applications in which the number of clusters is unknown a priori). In other words, the number of the patterns of daily behaviour can be increased if the number of input data increases. In more detail, DPM can either assign new data points to the learnt patterns of behaviours with the probability that each pattern generates the data or define a new pattern if the new data points are too different from the existing patterns of behaviour. Their work assumes that the daily going-out behaviour of the user is completed in a 24-hour cycle, and each behaviour of a day belongs to a certain category (e.g., going out for dinner in weekends, watching TV at home every evening). In this model, the Bernoulli distribution is used as a base distribution to describe the state of a user at certain time (that is away from home or not). Hence, the truncated stick-breaking process is applied to estimate the weights for each category. Once given a new observation of one day, the method can estimate to which category (daily behaviour) the day belongs.

Although these techniques are efficient at predicting a single user's behaviour, they do not address the challenges of the inter–dependency between different sequences of data (i.e., history of appliance usage) (Requirement 2). In addition, these algorithms can only predict within the temporal scope of one day, given initial observations of the same day, while in our scenario we need to forecast electricity consumption at least one day ahead. Moreover, we use DPM as a benchmark for our evaluation in Chapter 6. We will also benchmark our algorithms with an extension of the DPM model that is capable modelling inter-dependencies between appliances. More importantly, we propose our algorithm, GM-PMA, to take into account the human's routine as well as the inter-dependencies between appliances to enhance the usage prediction accuracy (Requirement 3). By so doing, we will show that GM-PMA efficiently predicts appliance usage in Chapter 6.

Before moving to describe our models, in the following section, we provide a brief background on probabilistic graphical models that we will use in the following chapters.

2.4 Probabilistic Graphical Models

Probabilistic graphical models are a powerful formalism for multivariate statistical modelling by bringing together graph theory and probability theory. They provide a simple way to visualise the structure of probabilistic models, and provide insights into the properties of the model such as conditional dependence between random variables. Given this, we use probabilistic graphical models to develop algorithms that satisfy all the requirements set out in Chapter 5. This section provides a background on probabilistic graphical models that we will use throughout this thesis. In particular, we describe Hidden Markov Models in Section 2.4.1, and then, we discuss some methods for approximate inference in graphical models in Section 2.4.2

2.4.1 Hidden Markov Models

A hidden Markov model (HMM) is an example of a graphical model that has been widely applied in many real-world applications.[10] In particular, an HMM is a graphical model in the form of a chain (see Figure 2.3). It defines a probability distribution in sequential data where sequences of observations $Y = \{Y_1, \ldots, Y_I, \ldots, Y_T\}$ can be invoking another sequence of unobserved (hidden) discrete state variables $X = \{X_1, \ldots, X_I, \ldots, X_T\}$. An important property of HMM is that the sequence of hidden states has Markov dynamics that the conditional probability of node X_i, given its immediate predecessor X_{i-1}, is independent of all other preceding variables. In addition, the model is defined in terms of three sets of parameters. First, the initial state X_1 is given a probability distribution $\pi = P(X_1)$. Second, the matrix of transition probabilities from state $i-1$ to state i can be represented by matrix $A = P(X_i|X_{i-1})$. This matrix is invariant across time. Third, the emission matrix for nodes Y_i can be represented by matrix $B = P(Y_i|X_i)$.

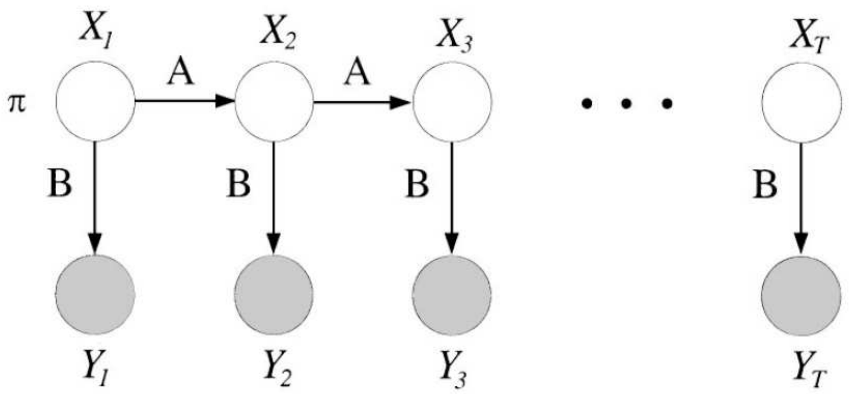

Figure 2.3: A HMM represented as a graphical model. The left-to-right spatial dimension represents time. The output nodes Y_i are observation nodes, and the state nodes X_i are hidden.

In the context of learning in HMM, the output nodes are treated as observation (evidence) nodes while the state nodes are treated as hidden nodes. The state–of–the–art algorithm for maximum likelihood estimation is the expectation-maximisation (EM) algorithm. However, approximate inference methods are more commonly used to reduce computational costs and to achieve fully Bayesian inference. In the next section, we discuss approximate inference techniques in more detail.

[10]We use HMM in Chapter 4.

2.4.2 Approximate Inference

Approximate inference methods are often applied to perform inference over probabilistic graphical models due to the expensive computational complexity of exact multivariate inference. In this section, we discuss two common approximate inference methods, namely Gibbs sampling in Section 2.4.2.1, and Variational Bayes in Section 2.4.2.2.

2.4.2.1 Gibbs Sampling

Algorithm 1 Pseudocode for Collaped Gibbs sampling.

```
initialise all variables;
while number of iterations is less than the maximum iteration do
  for each variable do
    sample new value conditioned on all variables;
  end for
  increment number of iterations;
end while
```

Algorithm 2 Pseudocode for Blocked Gibbs sampling.

```
initialise all variables;
while number of iterations is less than the maximum iteration do
  for group of variables do
    sample new value from the joint distribution of the group conditioned on all
    variables;
  end for
  increment number of iterations;
end while
```

Algorithm 3 Pseudocode for Variational Bayes.

```
initialise variational distribution of all variables;
while increase in lower bound is greater than threshold and number of iterations is
less than maximum do
  for each variables do
    update variable's variational distribution;
  end for
  compute lower bound on joint likelihood;
  increment number of iterations;
end while
```

Gibbs sampling is a Markov chain Monte Carlo (MCMC) algorithm which can be used to approximate samples from a (marginal) distribution indirectly, without having to calculate the density (Casellaa and Georgeb, 1992). In particular, there are two main Gibbs sampling techniques: i) Collapsed Gibbs sampling, and ii) Blocked Gibbs sampling. The Collapsed Gibbs sampling algorithm (as shown by Algorithm 1) involves replacing the value of one of the variables by a value drawn from the distribution of that variable conditioned on the values of the remaining variables. However, this technique considers one variable at a time, and thus there might be strong dependencies between successive samples. As a result, it is necessary to down-sample the data in order to ensure independence between samples. To increase independent successive samples, the Blocked Gibbs sampling algorithm (as shown by Algorithm 2) chooses groups (or blocks) of variables, rather than individual variables, then successively samples from the joint distribution from the variables in each block conditioned on the remaining variables (Jensen et al., 1995).

Gibbs sampling is widely used for inference. However, it might require a large number of iterations until convergence to the stationary distribution. To address this, the first specified number of samples are normally discarded. Due to the large number of iterations and each variable being repeatedly sampled in each iteration, it may be not computationally efficient. In the next section, we discuss Variational Bayes, an alternative technique for the more computationally efficient MCMC algorithm.

2.4.2.2 Variational Bayes

Variational Bayes (as shown by Algorithm 3) is used for approximate inference and works by iteratively updating the variational distribution of all variables in the model, and thus yields a lower bound on the model likelihood function (Blei and Jordan, 2006). The algorithm stops once either convergence is achieved or the maximum number of iterations has been reached.

Variational Bayes is faster in convergence than Gibbs sampling because bounds or functions are found over individual variables or groups of variables within a model at each iteration, rather than sampling from a conditional distribution.

In the following section, we discuss and summarise this chapter.

2.5 Summary

In this chapter, we have described the vision of the smart grid, discussed smart meters and the use of DSM techniques. We detailed the challenge for standard DSM to manage electricity demand on a large scale (thousands to millions smart meters) and discussed how agent-based approaches could help meet this challenge. We also discussed how recent agent-based DSM approaches have simplified the user's settings (Requirement 1) and have not considered the dependencies between appliances (Requirement 2) and hence are unlikely to be widely adopted. Moreover we identified the need to predict appliance usage in order to maintain user comfort.

We then discussed existing approaches for modelling and predicting inter-dependencies in time-series data. We described two popular approaches for modelling dependencies between events in temporal data: i) Temporal Point Process, ii) a rule-based method based on Frequent Episode Discovery. In particular, we use the state–of–the–art algorithms: PCIM (for a temporal point process approach), and EGH (a rule-based method) as benchmarks to compare the appliance usage prediction accuracy in Chapter 6. However, these approaches do not take into account human routine (Requirement 1), and thus we expect them to fail in predicting appliance usage (Requirement 3). Indeed we demonstrate this later in Chapter 6.

Building upon this, we discussed existing work for modelling and predicting human routine in some different applications such as location prediction, mobility prediction, and activity recognition. We use non-parametric method, DPM, as our benchmarks for predicting human routine. One of the popular models for detecting human routine is DPM. However, DPM only deals with one single appliance sequence, and does not take into account the dependencies between multi-appliances (Requirement 2). Therefore, the usage prediction accuracy will not be accurate (Requirement 3).

Lastly, we provided background of probabilistic graphical models such as HMMs which will be used in Chapter 4. We also discussed two main approximate inference methods: Gibbs sampling and variational bayes that will be discussed and used in Chapter 5.

To develop algorithms that address the research requirements, more importantly, we first extend the rule-based method by considering capturing the human's routine, and thus propose a rule-based algorithm, EGH-H (see Chapter 4). The EGH-H algorithm is mainly designed for scenarios in which notifications are needed in a short time. Again, we will demonstrate its effectiveness in Chapter 6. To model the more accurate appliance usage prediction, we propose our graphical model based, GM-PMA (see in Chapter 5) that also captures both human routine and the inter-dependencies between appliances. We will also show that GM-PMA dominates other benchmarks for accurate prediction in Chapter 6. Both algorithms, GM-PMA and EGH-H, takes into account the cyclic behaviour of users following the work of González et al. (2008). They have found strong periodicities in human location behaviour, and from common sense about how people use appliances in the home, it is reasonable to assume that such periodicities are true for appliance usage. Hence, we assume that human behaviour in terms of appliance usage forms cycles of weekly periods. We will show later about how this assumption is validated by the model performs well empirically (see Chapter 6).

In order to objectively compare the accuracy of various algorithms for predicting appliance usage, it is necessary to test them on real–world dataset. The following chapter describes the appliance usage prediction model, and discuss real–world dataset that we will use for evaluation.

Chapter 3

Appliance Usage Prediction Model and Datasets

This chapter describes the model of appliance usage prediction problem and the real–world datasets that we use for evaluating our proposed algorithms in the next few chapters. We start by modelling the appliance usage in the home, and discuss inter–dependencies between the appliances as well as the cycle of human routine for using the appliances in Section 3.1. Next, we provide two types of real–world datasets that will be used for evaluating the performances of the algorithms later in Chapter 6: i) expensive deployment for data collection, but low level of uncertainty; ii) cheap deployment for data collection, but high level of uncertainty. We describe the preparation of the datasets in more detail in Section 3.2. Finally, Section 3.3 summarises the findings of this chapter.

3.1 Model Description

Our goal is to generate time–specific predictions of appliance usage based on historical behaviour. In more detail, given a set of training data of past behaviour consisting of a time context indicating the day of the week and appliance usage, we wish to predict *which* appliances are likely to be used, and *when* they are likely to be used during the day. Assume that we have a finite set of appliance usage events, where different types of appliances are distinguished by $l \in L$. We consider a typical domestic profile that spans the usage over a day divided in T slots represents by a set of time slots $t \in T$ (e.g., if T = 24, the day is divided into one-hourly slots). We are concerned with modelling discrete binary information, $x_{n,l,t} \in \{0, 1\}$, indicating whether appliance l will be used on day n at time t. Let $x_{n,l} = \langle x_{n,l,1}, \ldots, x_{n,l,T} \rangle$ denote the observations of appliance l on the n^{th} day. Then, x_n is the observation of a subset appliances L on the n^{th} day. In probabilistic terms, we need to find the conditional probability $p(x_{n,l,t}|\mathbf{X}, n, l, t)$, where $\mathbf{X} = \{x_{1,l}, \ldots, x_{n-1,l}\}$ represents the history of the use of appliances. To solve this problem, we need to take into account two main dependencies that underlie a consumer's activities:

- **Time dependencies**: consumers can undertake their activities at different times that satisfy their needs and daily routine. For example, if one uses a washing machine at time t, it is less likely one will use it again at time $(t + 1)$ as one wash implies a reduction in the number of dirty clothes.[1]
- **Activity inter–dependencies**: some types of activities may depend on other activities. For example, while cooking, one might need to use the oven and microwave, then one might need to use the dishwasher to wash the dirty dishes. Therefore, the activities of using the oven, microwave, and dishwasher are inter–dependent.

In Chapter 4 and 5, we develop algorithms that take into account these dependencies of appliance usage. In the next section, we analyse the set of real–world datasets that we use to evaluate our algorithms in Chapter 6.

3.2 Real–world Datasets

In this section, we make use of two datasets that are collected from a field trial of energy feedback systems and are used in our experiments (Chapter 6) to evaluate our algorithms and the benchmark approaches. In particular, we use the REDD dataset (Kolter and Johnson, 2011) and the FigureEnergy dataset (Costanza et al., 2012). The REDD dataset is produced using highly intrusive methods that monitor appliance–level energy usage at high granularity and with high accuracy. Instead, the FigureEnergy dataset is collected by using non–intrusive methods (house-level monitoring), but involved high uncertainty (as appliance usage events were manually tagged by users who may have either mistaken the information of the event or forgotten to tag the labels). We describe the two datasets in more detail in Section 3.2.1 and Section 3.2.2 respectively.

3.2.1 The REDD dataset

The REDD dataset includes six different houses. These houses have been monitored for approximately 35 days with sub-meters installed on multiple relevant electrical home appliances. The raw data in the REDD dataset is the power consumption for the specific devices every 3 seconds. As we are interested in *when* and *which* appliances are being used during particular days. Thus, we need to prepare the data before using it. To do

[1]This may not be true of course for some users and this is what the algorithm is meant to detect as well.

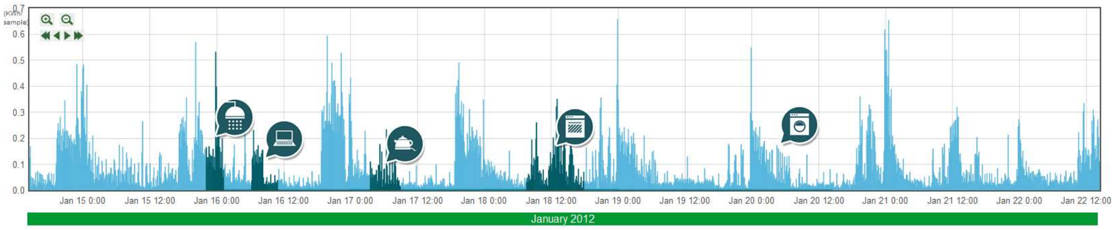

Figure 3.1: An example of using annotation in the FE application. The graph denotes power consumptions and the highlighted segments are the annotations of the user.

so, we converted the raw data of power consumption into a list of cyclic on–off events (i.e., a list of tuple $\langle$ appliance name, starting time, end time $\rangle$), and use these lists to test our prediction performance. The appliance usage detection method is as follows:

- We set a threshold of power consumption (typically 55W) to determine the periods that the appliances turned on. We store all these segments of durations when the appliances are turned on.
- We set a *gap allowance parameter* for two consecutive segments. If the gap between these two consecutive segments is greater than the gap allowance, we connect these two segments together, and considered them as one segment.
- We select a *noise removal parameter* to filter the noise of the data. All the segments that have values lower than the noise parameter, are removed.

Figure 3.2 shows an example of converting raw data into a list of on–off events. The *gap allowance parameters* and *noise removal parameter* are adjusted to adapt to the behaviour of the appliances. For example, the period of *dishwasher* cycle is typically over 30 minutes. Hence we set the noise parameters for the full cycle of using the dishwasher is up to 30 minutes. In addition, the power consumption of the dishwasher is controlled by the built-in temperature sensor in the dishwasher. Given this, the power consumption consumed for the dishwasher fluctuates. However, the gap between the two consecutive periods where the power consumption over 55W is less than 10 minutes. We observed that there were 3 houses that do not have enough data to judge the performance of the prediction. Hence, we only carry out our tests on data from the other 3 houses. Moreover, for the FE dataset, we use the user's labelling (i.e., appliance usages) directly. We describe the FE system in the following section.

3.2.2 The FigureEnergy Dataset

FigureEnergy which is a web–based application designed for appliance usage labelling to allow users to identify and label the activities. In particular, the dataset included 13 participating homes. Each household was given an off–the–shelf energy monitoring device,[2] which integrated into the user's home and transferred data into the application's server over the internet. Users then could observe their aggregated energy consumption from their web browser using FigureEnergy. This application allows users to identify and label individual activities as follows. By clicking on the graph with their mouse and dragging, users can select a segment and fill information about the activities that they spent. There is a preset list of labels that users can choose from for their activities. The labels are also displayed on the aggregate energy consumption graph to show users the results that they have annotated (see Figure 3.1). An example of the collected data can be seen in Table 3.1. The labels of appliance usage are being used directly for evaluating the prediction performance. Please note that users can tag the wrong information (e.g., incorrect type of appliance, and time), and forget to tag their past activities (due to short term memory retrieve). This raises the uncertainty for the FE dataset. Now, we summary this chapter in the following section.

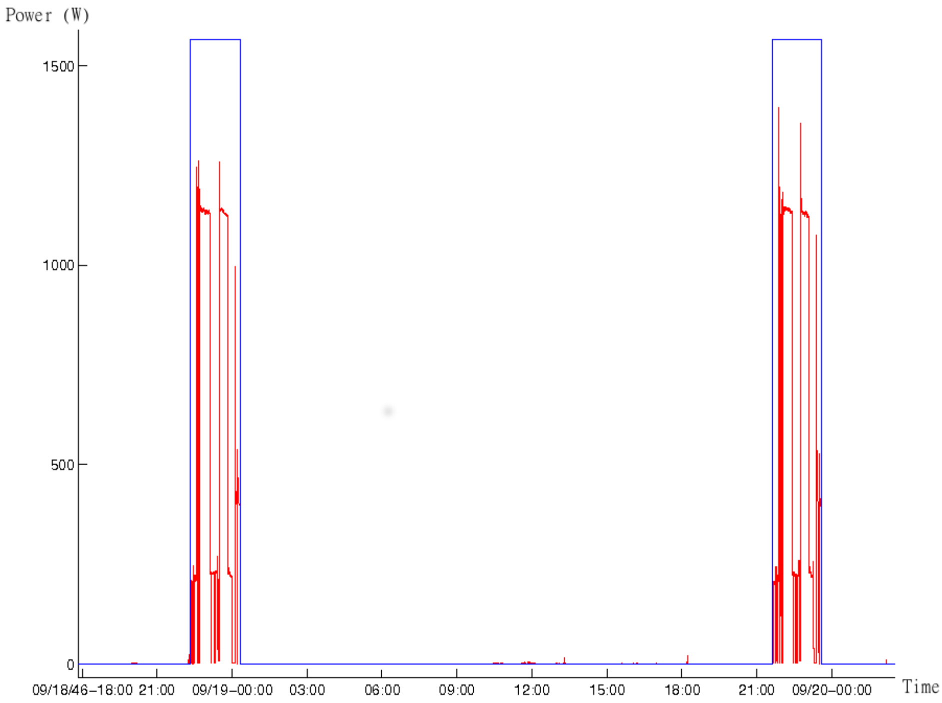

Figure 3.2: An example of how to convert raw data of power consumption into a list of cyclic on–off events.

3.3 Summary

In this chapter, we described the model for the appliance usage prediction problem more formally. We also identified two main dependencies that underlie the use of appliances

[2] AlertMe

Type	**Start time (time unit)**	**End time (time unit)**	**Energy usage (J)**	**Baseline (J)**
oven	2011-08-31 18:58:27	2011-08-31 19:42:47	1.562	0.069
kettle	2011-09-01 07:21:05	2011-09-01 07:26:17	0.094	0.007
shower	2011-09-01 08:12:45	2011-09-01 08:21:08	0.102	0.0144
tv	2011-09-01 17:54:16	2011-09-01 19:18:20	0.3902	0.151
stove	2011-09-01 19:18:20	2011-09-01 19:41:16	0.585	0.0396

Table 3.1: An example of data collected from FigureEnergy for user 32.

in the home: i) time dependencies, ii) appliance inter-dependencies. Then we described two real–world datasets: the REDD dataset, and the FE dataset. We discussed data preparation techniques for the REDD dataset before evaluating the prediction performance in Chapter 6. Now, we turn to describe our first proposed algorithm, EGH-H, which is designed for scenarios in which the notifications are quickly generated, in the next chapter.

Chapter 4

The EGH-H Algorithm

This chapter describes the EGH-H algorithm that was developed for scenarios in which notifications need to be given at short notice. In general, EGH-H builds upon the EGH algorithm proposed by Laxman et al. (2005). As EGH is not designed for detecting human activities, we adapt the algorithm to suit our settings by fitting it with a model of the periodic nature of human routine. We first detail the training phase of our approach in Section 4.1, where we use a set of training data to build up a dependency model for the correlations between the usage of appliances. We then describe a human routine model in Section 4.2. Based on these models, we then construct a mixture model of the significant episodes (i.e., sets of possible inter–dependency rules) in order to calculate the probability of appliances being used in Section 4.3. Section 4.4 describes the appliance usage prediction using the constructed mixture model. Finally, Section 4.5 summarises the findings of this chapter.

4.1 The Inter–Dependency Model

To build the inter–dependency model, we first assume that the inter–dependency between the usage of different appliances follows some probabilistic patterns. For example, a typical pattern can be the following: it is likely that the dryer is also used after the usage of the washing machine, or the use of the microwave follows the use of the oven. In our model we denote these patterns as *episodes* (i.e., a sequence of appliance usage).

To evaluate the likelihood of an episode, we calculate its probability of occurrence, given the history of appliance usage. To do so, we use the EGH approach, which assigns a discrete hidden Markov model (HMM) to the corresponding episode (note that the EGH's background can be seen in more detail in Section 2.2.2.4). In particular, suppose that episode φ consists of p usage of the appliances $(x_{1,l_1,t_1}, x_{2,l_2,t_2}, \dots, x_{p,l_p,t_p})$ that are in temporal order (i.e., the appliance usage x_{p,l_p,t_p} occurred before the next appliance usage

$x_{p+1,l_{p+1},t_{p+1}}$). EGH assigns a special discrete HMM $\Lambda_\varphi = (S, \Delta_\varphi, \eta_\varphi)$ to φ such that $S = \{1, \ldots, p, p+1, \ldots, 2p\}$, denotes the state space (includes two states: the episode states $= \{1, \ldots, p\}$, and the noise states $= \{p{+}1, \ldots, 2p\}$), $\Delta_\varphi = (x_{1,l_1,t_1}, x_{2,l_1,t_2}, \ldots, x_{p,l_p,t_p})$ denotes the appliance usage, and η_φ is the noise parameter. More particularly, the noise parameter η_φ is defined as follows:

$$\eta_\varphi = \begin{cases} \frac{T_\varphi - p f_\varphi}{T_\varphi} & if\ \eta_\varphi \le \frac{\hat{M}}{\hat{M}+1} \\ 1 & otherwise. \end{cases} \tag{4.1}$$

where T_φ is the total number of appliance usage in the training dataset and f_φ is the number of times the episode φ occurs in the training dataset, and $\hat{M}$ is a size of the episode . The intuition behind the use of Λ_φ is that it represents a Markov model of a sequences activities that contains the corresponding episode (see Laxman et al. (2005) for more details).

Now, we calculate the corresponding frequency f_φ of each possible episode φ within the dataset (i.e., the number of times the episode occurs in a non–overlapping way). Hereafter, we only consider those episodes that have a frequency f_φ higher than $\frac{T_\varphi}{p(\varphi)\hat{M}}$, where $p(\varphi)$ is the number of usage of the appliances within φ. We denote these as *significant episodes*. The reason we focus on these episodes (and thus, ignore the rest) is that the others are unlikely to occur given the training data set. These significant episodes can be regarded as rules that model the inter–dependency between the occurrence of different appliance usage.

4.2 The Human Routine Model

By building up the set of significant episodes, we can then predict the occurrence of appliance usage within the next time step (e.g., hour, or day) by analysing whether they can be part of a significant episode. Also, the prediction model can be repeated to predict for a future day, or week ahead. However, as the number of significant episodes can be an exponential in the size of the training dataset, EGH may have high computational costs. In addition, EGH may overestimate the occurrence of usage of the appliances, due to redundant episodes. In particular, due to the cyclic nature of human routines, a sequence of appliance usage that consists of two non–overlapping, but identical, episodes can also be regarded as a significant episode. This can lead the inaccurate estimation of the probability of the occurrence of the appliance usage.

To address these challenges, we reduce the set of potential significant episodes by exploiting the cyclic features of human everyday routine. In particular, we assume that human behaviour in home energy usage follows a weekly cycle. Thus, if the goal is to predict whether a target appliance l occurs on the day of the week q_n, we focus on learning the user's behaviour for the use of appliance l on the same day of the week q_n in the past by capturing the appliance dependency and cyclic human behaviour that triggered the actual use of the appliance l during the day. More formally, let C denote the number of occurrences of the target appliance l on the specific day of the week q_n in the activity usage history X. Thus, for each label l and the prediction day of the week q_n, from the original training dataset D, we extract a training set $D_l = \{X_i\}_{i=1}^{C}$, where $X_i = < x_{i-w,l,t}, \ldots, x_{i-1,l,t} >$ is the preceding window of activities from $x_{i,l,t}$ that immediately preceded the i^{th} occurrence of appliance l on the day of the week q_n in the historical observation of appliances X. We consider a weekly cycle in our experiment (i.e., $w = 7$).[1]

Given this reduced training dataset D_l, we then use the EGH approach to identify the significant episodes. The intuition behind this technique can be described as follows. We assume that the appliance usage events are typically influenced only by the use of appliances within a week (i.e., older activities do not have any effect on them). By so doing, we can reduce the computational costs and also improve the quality of prediction (as we will demonstrate later in Chapter 6).

4.3 The Mixture Model

Given the episode reduction using the human routine model, we now turn to the discussion of how to use these episodes to predict future appliance usage. To do so, we first analyse the joint influence of these episodes on the probability of occurrence of a single future activity. Suppose that for a given training data set $D_l = \{X_i\}_{i=1}^{C}$, we have calculated a set of significant episodes, denoted as $F^s = \{\varphi_1, \ldots, \varphi_J\}$, and each HMM Λ_{φ_j} of episode φ_j. Now, to predict the usage of appliance l in the next time step t, we use these episodes in order to calculate the probability of occurrence. To do so, we calculate the probability that l is a part of a significant episode. However, as an episode typically has a certain positive probability of indicating the occurrence of l, we have to take into account all of HMMs. To model the effect of this joint influence, we compute a mixture model Λ_l (i.e., a combination of probabilistic processes) of the significant episodes' HMMs. This mixture model can then be used to predict the future occurrences of the target appliance l. In what follows, we first build the aforementioned mixture model and then demonstrate how to predict future activities.

Now, the likelihood function of the training dataset D under a mixture model Λ_l can be written as follows:

$$P[D|\Lambda_l] = \prod_{i=1}^{C} P[X_i|\Lambda_l] = \prod_{i=1}^{C} \left(\sum_{j=1}^{J} \theta_j P[X_i|\Lambda_{\varphi_j}] \right) \tag{4.2}$$

[1]Note that we can easily change to other period lengths by updating the parameter of cycle length.

where θ_j, $j = 1..J$ are the mixture coefficients of Λ_l (with $\theta_j \in [0,1]$ for all j, and $\sum_{j=1}^{J} \theta_j = 1$). Recall that each HMM Λ_{φ_j} is fully characterised by the significant episode φ_j and its noise parameter η_{φ_j}. Given this, the likelihood of the activity sequence X_i, given the HMMs $\{\Lambda_{\varphi_j}\}_{j=1}^{J}$, is computed by approximating the likelihood along the corresponding most likely state sequence:

$$P[X_i|\Lambda_{\varphi_j}] = \left(\frac{\eta_{\varphi_j}}{M}\right)^{|X_i|} \left(\frac{1-\eta_{\varphi_j}}{M}\right)^{|\varphi_j| f_{\varphi_j(X_i)}} \tag{4.3}$$

where $|X_i|$ denotes the length of sequence, X_i, $f_{\varphi_j}(X_i)$ denotes the non-overlapping occurrences-based frequency of φ_j in the sequence X_i, and $|\varphi_j|$ denotes the size of the episode φ_j.

We use the Expectation Maximisation (EM) algorithm to estimate the set of mixture coefficients of the mixture model Λ_l. In particular, the algorithm is initialised with the current guess for the mixture coefficients, denoted by $\Theta^g = \{\theta_1^g, \ldots, \theta_J^g\}$. These mixture coefficient values are initially set to be uniform, that is, $\theta_j^g = \frac{1}{J}$ for every $j \in J$. We then use these current guesses to update the mixture coefficients as follows. Let Θ^{new} $= \{\theta_1^{new}, \ldots, \theta_J^{new}\}$ denote the new values of these coefficients. Given this, we have:

$$\theta_q^{new} = \frac{1}{C} \sum_{i=1}^{C} P[q|X_i, \Theta^g] \tag{4.4}$$

where $q = 1..J$. Let $P[q|X_i, \Theta^g]$ denote the posterior probability for the q^{th} mixture component, with respect to the window $X_i \in D_l$, which can be computed using Bayes' Rule:

$$P[l|X_i, \Theta^g] = \frac{\theta_l^g P[X_i|\Lambda_l]}{\sum_{j=1}^{J} \theta_j^g P[X_i|\Lambda_{\varphi_j}]} \tag{4.5}$$

The new set of mixture coefficients Θ^{new} is then used as the current set of guesses (i.e., Θ^g) of the mixture coefficients. The process is repeated until the coefficients converge.

4.4 Appliance Usage Prediction

Given the mixture model $\Lambda_l = \{(\varphi_j, \theta_j)\}_{j=1,\ldots,J}$, we now turn to the prediction phase of our approach. Let n denote the current day. For the set of target appliances $l \in \mathrm{L}$, we want to predict their usage in the next day, $n+1$. As we are mainly interested in the recent usage of appliances of the users, therefore, we construct the weekly period $[n-7, n]$ of the usage of appliances. The recent list of usage of appliances can be written as $X_n = [x_{n-7,l,t}, \ldots, x_{n-1,l,t}]$. We then estimate the likelihood of this list of recent appliance usage events, given the mixture model, Λ_l, that is obtained from the training phase. The algorithm determines the usage of the target appliance at time step $(n+1)$ based on the value of the threshold. In particular, if the probability of the window under the mixture model is greater or equal than the prediction threshold, the algorithm predicts that the target appliance will be used at the next time step $(n+1)$. Otherwise, if the probability of the window under the mixture model is less than the threshold value, the algorithm predicts that the target appliance will not be used at the next time step $(n+1)$.

We will show that EGH-H can perform the prediction quickly in Chapter 6. The algorithm is best suited for scenarios that notifications need to be given in a short time. However, EGH-H may not be the best algorithm for such scenarios that prefer the more accurate prediction rather than the runtime. To improve the prediction accuracy, we develop an alternative algorithm, GM-PMA. We describe it in Chapter 5. Next, we summarise this chapter in the following section.

4.5 Summary

In this chapter, we presented the EGH-H algorithm that can be used to forecast the appliance usage in limited time. To capture the inter–dependencies between appliances, we defined the inter–dependency model that can identify the significant episodes (i.e., patterns). We also defined the human routine model that allows EGH-H algorithm to find these significant episodes based on the construction of the (weekly) cyclic nature of human routine over the historical data. By so doing, EGH-H can improve the prediction accuracy as well as the computational cost (as we will show in Chapter 6). Then, for each significant episode, we applied the discrete HMM to model the episode's structure. We took the effect of the joint influences of all episodes by modelling the mixture model (i.e., all HMMs). Hence, we used this mixture model to predict for the future use of appliances.

We will show that EGH-H algorithm can deliver reasonable prediction accuracy quickly (i.e., seconds). It is best suitable for scenarios where consumers need to be notified in a short time. To improve the prediction accuracy, we provide an alternative solution by developing our graphical model based algorithm (GM-PMA). We elaborate on our GM-PMA algorithm in the next chapter.

Chapter 5

The GM-PMA Algorithm

In this chapter we propose GM-PMA algorithm, a graphical model based method for the prediction of appliance usage. In particular, we present our model based on appliance interdependency in Section 5.1. We then describe the algorithm for model inference (based on training data) in Section 5.2, and finish with the equations required to perform prediction with this model in Section 5.3. Finally, we summarise the findings of this chapter in Section 5.4.

5.1 The Inter–Dependency Clustering Model

A key assumption we make in GM-PMA is that such multi–appliance usage behaviour comes in blocks of fixed size, where each block represents a single day of activity. This approach has been effective in related areas of human presence prediction (Tominaga et al., 2012) and derives from the periodic features of human behaviour that have been widely observed in empirical data (González et al., 2008). Our extension is to also consider the conditional dependencies between day blocks of behaviour for a single household (multiple day dependencies) and within day blocks (intra–day dependencies).

In general, we trade–off complicated dependencies between day blocks in favour of the assumption that each day of behaviour is independent of any other day given the assignment of days to discrete classes of behaviour. Since supervised labels of these assignments are unavailable, we consider them to be latent random variables in our model. These latent classes compactly represent sets of behaviours that we call *day types*. As we will show, it is possible to infer the nature of these day types in an unsupervised way. Intuitively, day types can be understood as representing, for example, working days, weekend days, or family visiting days. Day types can be considered latent classes of a mixture model, albeit with a non-standard structure to the observations which we detail next.

Within day blocks, we now discuss the random variables controlling observations within each day block (i.e., intra–day dependencies). Since we are interested in predicting far ahead in time, (i.e., the next day or next several days of appliance usage), there is little advantage in making behaviour at one time of the day dependent on behaviour at another time, because we have already made the assumption that each day's behaviour is generated from a hidden day type class. In more detail, because day types are never directly observed (i.e., the random variable indicating latent assignment is never instantiated), we already achieve dependencies between appliance usage at different times of the day. This can be intuitively understood as a flow of information between the random variables indicating appliance usage at different times of the day, all flowing *via* the latent assignment of that day's behaviour. We may make a similar argument for dependencies between appliances (e.g., between the oven and the kettle, or between the television and lighting). In summary, we achieve the desired dependencies between appliance usages whilst simultaneously using the fast and well–established machinery of mixture modelling, by taking advantage of the periodicities of routine behaviour and the fact that appliance use is explained by a set of uninstantiated day types (e.g., weekend, holiday).

We now formalise these assumptions into a full Bayesian model of the appliance usage in a single home, that avoids problems of over–fitting which can affect models involving large numbers of parameters relative to the number of observations. This requires the specification of two main components to the model (along with their respective parameters): (i) the likelihood function of appliance usage and day of the week observations, and, (ii) the prior distribution over latent day types. The observations and parameters to this model are summarised in Figure 5.1. The dependencies between parameters is represented by directed arrows. In this graph, the observation $\mathbf{x}_{n,l}$ $(n \in N)$ depends on the day types μ_k $(k \in K)$, the indicator parameter $\mathbf{z}_n$ (generated from the prior distribution π_k) that indicates which day type the observation $\mathbf{x}_n$ belongs to, and the day of the week $\mathbf{q}_n$ $(\mathbf{q}_n \in W)$ is parameterised by multinomial distribution σ_k with prior Dirichlet distribution γ. In what follows, we elaborate on our graphical model by showing how we form the likelihood functions of the appliance behaviour and how we employ the Dirichlet Process Mixture to estimate day type classes.

5.1.1 Likelihood Functions

Starting with the likelihood functions, the behaviour for each appliance throughout a day can be represented by a Bernoulli distribution:

$$p(\mathbf{x}_{n,l}|\mu) = \mu^{\mathbf{x}_{n,l,t}}(1-\mu)^{1-\mathbf{x}_{n,l,t}} \tag{5.1}$$

where $\mathbf{x}_{n,l}$ be the observation of appliance l on n^{th} day, $\mu = \mu(t)(0 < \mu(t) < 1)$, and $\mathbf{x}_{n,l,t} = \{0, 1\}$ be the observation of appliance l on n^{th} day at time slot $t \in T$ of the

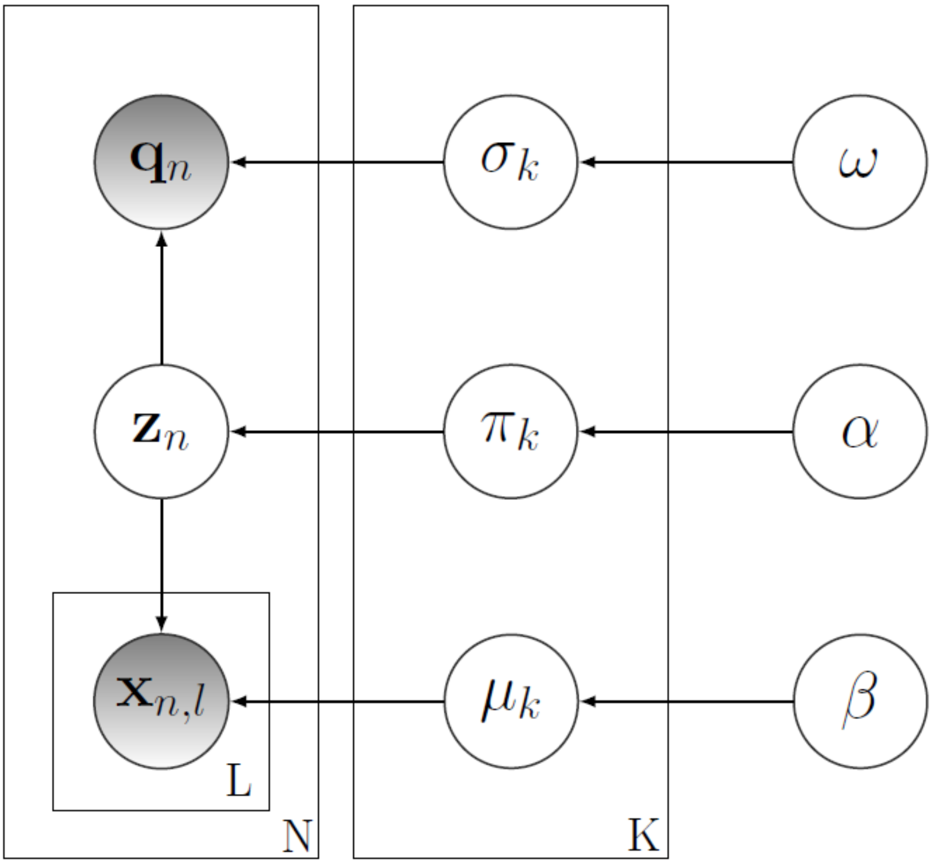

Figure 5.1: Graphical model for the usage of multi appliances in the home. Shaded nodes indicate observed information.

day (0: not being used, 1: being used). For example, Figure 5.2 shows the use of the washing machine on a specific day (i.e., $\mu_{n,l,t}$ for $0 \leq t \leq 24$). In this graph, we can see the washing machine is likely to be used around 18:00 with the probability is 0.9. The day type can be used to describe the behaviour of the appliance on a specific day. Let $k = 1, \ldots, K$ be the ID of the day types. Each day type has a sequence of parameter $\mu_k = (\mu_{k,l,1} \ldots \mu_{k,l,T})$, where $\mu_{k,l,t} \in [0, 1]$ represents for the probability of the appliance l belongs to the day type class k in the time of the day $t \in T$. Given this, the likelihood of the observation $\mathbf{x}_n$ is:

$$p(\mathbf{x}_n|\mu_k) = \prod_{l=1}^{L}\prod_{t=1}^{T}\mu_{k,l,t}^{x_{n,l,t}}(1-\mu_{k,l,t})^{1-x_{n,l,t}} \tag{5.2}$$

The beta distribution is used as a conjugate prior for the parameter $\mu_{k,l,t}$. That is:

$$\mu_{k,l,t} \sim \mathcal{B}(\mu_{k,l,t}|\beta_1, \beta_2) \propto \mu_{k,l,t}^{\beta_1-1}(1-\mu_{k,l,t})^{\beta_2-1} \tag{5.3}$$

where β_1 and β_2 are preset hyperparameters. Furthermore, we exploit the cyclic features of human everyday routine. In particular, we assume that human behaviour in term of

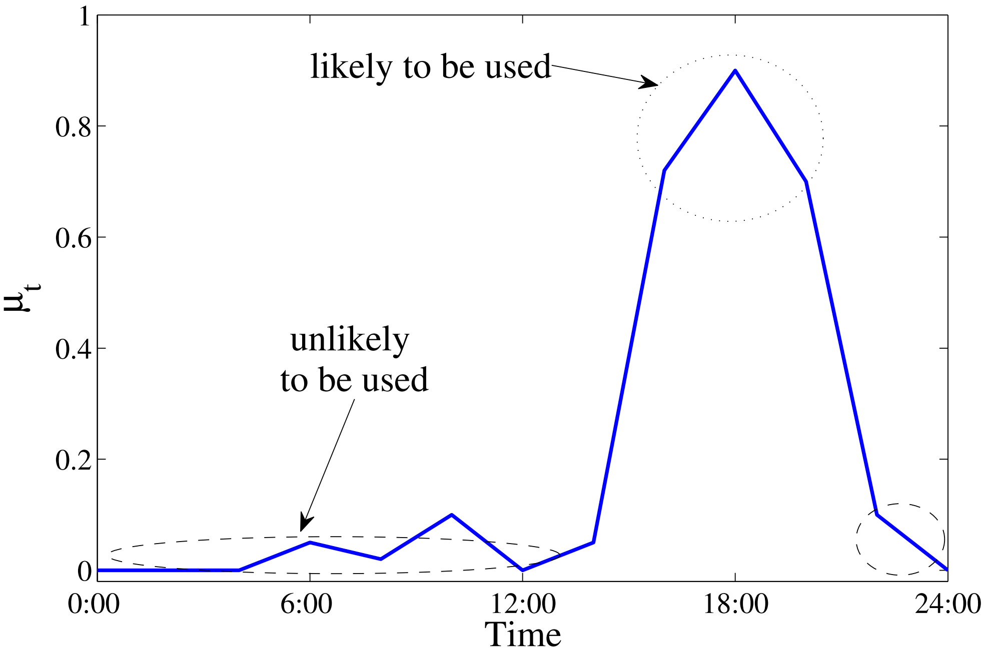

Figure 5.2: An example of the washing machine usage throughout a day.

appliance usage follows a weekly cycle (as in the case of EGH-H). Thus, to increase the accuracy of the prediction, we condition the indicator with the day of the week. More precisely, if the goal is to predict the activity usage profile on the next day $\mathbf{x}_{n+1}$, where $\mathbf{q}_{n+1}$ can be found as the day of the week, then we only consider the same day of the week $\mathbf{q}_{n+1}$ from the past to predict the activity profile on the $(n+1)^{th}$ day.[1] As we show later in Chapter 6, doing so can significantly improve prediction accuracy. Given $\sigma_{k,w}$ as the probability of the day type $k \in K$ belongs to the day of the week $w \in W$, the day of the week $\mathbf{q}_n$ is modelled using a multinomial distribution as $\mathbf{q}_n \sim \mathcal{M}(\mathbf{Z}, \sigma_{k,w})$ (see the dependency in Figure 5.1). In particular, we use conjugate Dirichlet distribution to model $\sigma_{k,w}$, which can be expanded as $\sigma_{k,w} \sim \mathrm{Dir}(\omega)$, where ω is a preset hyper parameter. We describe the use of Dirichlet Process Mixture in the next section.

5.1.2 Dirichlet Process Mixture

To denote that an appliance usage activity fits a given day type, we define an indicator $\{\mathbf{Z}\}_{n,k}$. In particular, $\{\mathbf{Z}\}_{n,k}$ indicates the observation on day $n \in N$ belongs to the day type ID $k \in K$ and is defined as follows:

$$\{\mathbf{Z}\}_{n,k} = \{\mathbf{z}_{n,k} | \mathbf{z}_{n,k} = \{0,1\}, \text{and} \sum_k \mathbf{z}_{n,k} = 1\} \tag{5.4}$$

[1]However, if more information is available, such as weather forecasts for the location of the home, or calendar information, then the model may treat these additional observations in a similar way to the day of the week observations, and get an even more accurate prediction of appliance usage.

As shown from the graphical model (see Figure 5.1), z_n is a random variable following multinomial distribution $\mathcal{M}(z_n|\pi)$. Thus, we employ the Dirichlet Process Mixture (DPM) to describe the infinite Dirichlet distribution with unknown component coefficient parameters as a prior distribution of day type classes. Previous work has shown that DPM parameterises the distribution of the size of the day type classes, and effectively estimates the number of day types as well as the parameters of the day types (Tominaga et al., 2012). The number of day types can in principle be infinite. However, in practice an upper bound K is set to a suitably large value (e.g., 50, 100). The method estimates the number of day types as $k \ll K$ that is guaranteed to be bigger than what would be expected in any given case. In particular, we apply DPM truncated stick–breaking process (Sethuraman, 1994) to approximate the infinite–dimensional Dirichlet distribution, which represents the beta distribution as a prior of each coefficient of a multinomial distribution. The coefficients π_k are calculated as:

$$\pi_k = \mathbf{v}_k \prod_{i=1}^{k-1} (1 - \mathbf{v}_i), \mathbf{v}_k \sim \mathcal{B}(\mathbf{v}_k|1, \alpha) \tag{5.5}$$

where α is a preset hyperparameter. Given this, we can now define the conjugate prior of the Dirichlet distribution for π as:

$$\pi \sim \mathrm{Dir}(\pi|\alpha) \propto \prod_k \pi_k^{\alpha-1} \tag{5.6}$$

where α is a preset hyperparameter. If $\mathbf{M} = (\mu_1, \ldots, \mu_K)$, and $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ be the parameter sequences of the day types, then we can define the likelihood of the appliance usage profile given all parameters is:

$$p(\mathbf{x}_n|\mathbf{Z}, \mathbf{M}) = \prod_{k=1}^{K}\prod_{l=1}^{L}\prod_{t=1}^{T} [\mu_{k,l,t}^{\mathbf{x}_{n,l,t}} (1 - \mu_{k,l,t})^{1-\mathbf{x}_{n,l,t}}]^{\mathbf{z}_{n,k}} \tag{5.7}$$

In the next section, we describe how these latent day type parameters are estimated.

5.2 Inference of Parameters and Cluster Number

It has been shown that among existing inference methods for implementing DPM, such as blocked Gibbs sampler (Ishwaran and James, 2001), variational Bayes (Blei and Jordan, 2006), collapsed Gibbs sampler (Maceachern, 1994), the blocked Gibbs sampler has higher probability to reach global optima than the others (Tominaga et al., 2012). Thus, we use the blocked Gibbs sampling within our paper. The Gibbs sampling process firstly initialises the parameters randomly. Then, it iteratively alternates resampling from the posterior distributions of the unknown random variables as follows:

$$\mathbf{V}, \mathbf{M}, \sigma \sim p(\mathbf{V}, \mathbf{M}, \sigma | \mathbf{X}, \mathbf{Z}) \tag{5.8}$$

$$\mathbf{Z} \sim p(\mathbf{Z} | \mathbf{X}, \mathbf{V}, \mathbf{M}, \sigma) \tag{5.9}$$

where $\mathbf{V} = \{v_1, \ldots, v_K\}$; $\sigma = \{\sigma_1, \ldots, \sigma_K\}$. Based on the aforementioned discussion, the posterior distributions can be calculated as follows:

$$\mathbf{v}_k \sim \mathcal{B}(\mathbf{v}_k | 1 + \sum_n z_{n,k}, \alpha + \sum_{i=k+1}^{K} \sum_n z_{n,i}) \tag{5.10}$$

$$\begin{aligned} \mu_{k,l,t} \sim \mathcal{B}(\mu_{k,l,t} | \quad & \beta_1 \quad + \sum_n \mathbf{x}_{n,l,t} \mathbf{z}_{n,k}, \\ & \beta_2 \quad + \sum_n (1 - \mathbf{x}_{n,l,t}) \mathbf{z}_{n,k}) \end{aligned} \tag{5.11}$$

$$\sigma_{k,w} \sim \mathrm{Dir}(\sigma_{k,w} | \omega + \sum_n \mathbf{q_n} z_{n,k}) \tag{5.12}$$

$$\mathbf{z}_n \sim \mathcal{M}(\mathbf{z}_n | \pi^*), \pi^* := \frac{\pi_k p(\mathbf{x}_n | \mu_k) p(q_n | \sigma_k)}{\sum_k \pi_k p(\mathbf{x}_n | \mu_k) p(q_n | \sigma_k)} \tag{5.13}$$

In Equation 5.10, we sample the weights $\mathbf{v_k}$, which are the beta random variables of the stick breaking construction of the DP, as per the standard stick-breaking construction (Sethuraman, 1994). The posterior distribution for weights can be directly calculated from the total counts of latent day type assignments and are added to the hyper parameters to find the current pseudo count of each beta distribution (Bishop, 2006). In Equation 5.11, a similar process is applied to the beta random variables μ indicating the probability of using appliances at all times of the day. Equation 5.12 defines how the day of the week probabilities for each day type are sampled from their posterior Dirichlet distribution. Finally, Equation 5.13 uses Bayes' theorem to incorporate the likelihood of appliance use observations and the prior distribution of day types to randomly sample the day assignments for each day block, $\mathbf{z_n}$. As per normal Gibbs sampling, we iterate through these sampling steps until convergence.

Next, we show how to use this model to perform prediction.

5.3 Appliance Usage Prediction

Given the historical observation for all appliances $\mathbf{X}$ and the parameters estimated from the training process, we now consider the prediction of the probability of all appliance usages for the next day $\mathbf{x}_{n+1,l} = \{\mathbf{x}_{n+1,l,1} \ldots \mathbf{x}_{n+1,l,T}\}$ (where $l \in L$). In this scenario, the day of week on the $(n+1)^{th}$ day is a known value, denoted as $\mathbf{q}_{n+1} \in W$. The prediction algorithm uses marginalization from unknown random variables. To ensure convergence, we use a standard technique known as 'burn-in' that discards the initial parameters samples (during Gibbs sampling) to ensure that we do not end up with inaccurate parameter estimates that were taken before convergence of the Markov chain in Monte Carlo. We found that convergence occurred for our model before 60 iterations. Thus, we discard the first 60 samples. Then, we downsample the remaining samples such that every 3^{rd} sample is retained. Finally, we obtain the mean probability for each appliance on the prediction day $(n+1)$. The likelihood for each appliance $l \in L$ can be expanded as:

$$\begin{aligned} & p(\mathbf{x}_{n+1,l}|\mathbf{q}_{n+1}, \mathbf{h}_N) = \\ & = 1/R \sum_{r=1}^{R} p(\mathbf{x}_{n+1}|\mathbf{q}_{n+1}^{(r)}, \pi^{(r)}, \sigma^{(r)}, \mu^{(r)}) \end{aligned} \tag{5.14}$$

where R is the number of samples obtained in the Gibbs sampling process. We marginalise over unknown day types for the day $(n+1)$, and take out the normalising constant $p(q_n)$, then $p(\mathbf{x}_{n+1}|\mathbf{q}_{n+1}, \pi, \sigma, \mu)$ can be computed as follows:

$$p(\mathbf{x}_{n+1}|\mathbf{q}_{n+1}, \pi, \sigma, \mu) \tag{5.15}$$

$$\sim \sum_{z_n} p(x_n|\mu, z_n)p(z_n|\pi)p(q_n|\sigma, z_n) \tag{5.16}$$

$$= \sum_{k=1}^{K} (\sigma_{k,\mathbf{q}_{n+1}} \pi_k \mu_{k,l}) \tag{5.17}$$

The algorithm determines the occurrence of the target appliance at time step $t \in T$ based on the value of the threshold in which we vary this threshold value from the minimum probability value to the maximum value from all prediction probability (see more detail in Chapter 6 while we evaluate the prediction performance of the algorithm). In particular, if the probability of the target activity at the specific time of the day is greater or equal than the prediction threshold, the algorithm predicts that the target activity will occur at the time of the day $t \in T$ at the next day $n+1$. Otherwise, if the probability of the target activity at the specific time of the day is less than the threshold value, the algorithm predicts that the target activity will not occur at that time of the day.

We show that GM-PMA outperforms other benchmarks in terms of prediction accuracy in the next chapter. We now summarise the findings of this chapter in the following section.

5.4 Summary

This chapter described GM-PMA algorithm that shows to be the best for prediction accuracy, though at the expose of computational time. At GM-PMA, we used Bernoulli distribution to model the behaviour of each appliance throughout the day (i.e., modelling the probability of when and which appliance is being used during the day). Then, we develop the Dirichlet Process Mixture based to identify the hidden classes that represents for a group of behaviours of the user (e.g., working days, weekend day). The GM-PMA also conditioned on the (weekly) cyclic nature of human routine, and thus enhances prediction accuracy. We then presented the use of blocked Gibbs sampling to infer the model's parameters and cluster numbers. We predict the future appliance usage by using the marginalisation from unknown random variables. Next, we evaluate the performance of our algorithms, GM-PMA, and EGH-H (Chapter 4) against other benchmarks in the following chapter.

Chapter 6

Empirical Evaluation

In this chapter, we evaluate GM-PMA, and EGH-H in predicting the usage of appliances during a day in the home. To do so, we first describe the settings of our experiments in Section 6.1. Then, we introduce a set of benchmark algorithms against which we compare our methods in Section 6.2. Section 6.3 provides evaluation metrics for our experiments. Section 6.4 describes synthetic data evaluation. Next, we empirically analyse the experiment results in Section 6.5. Finally, we summarise this chapter in Section 6.6.

6.1 Experiment Settings

In our model, we focus on predicting the use of the appliance at the certain time slot through a day (i.e., binary use). We consider the domestic profile where the usage over a day can be divided in two-hourly slots represented by a set of time slots $T = \{1, \ldots, 12\}$ because some appliances (e.g., washing machine) can be lasted for up to 2 hours.[1] In addition, for all houses in our datasets,[2] we use the first 75% as a training data set, and the remaining 25% as a test set (i.e., comparing the appliance usage prediction against the ground–truth dataset). Now, we describe the benchmark algorithms in the following section.

6.2 Benchmark Algorithms

Related work has typically focused on single user behaviour prediction and dependency model prediction for non–human data (see Chapter 2). Given this, we choose a number

[1]Note that we focus to answer the question whether or not the appliance is being used at the given time slot.
[2]The REDD dataset, and the FigureEnergy dataset.

of state–of–the–art methods from these domains to benchmark against. In particular, we compare our method against the following approaches:

- **The piece–wise constant conditional intensity model (PCIM)**: a state–of–the–art approach in predicting multiple–source web data where data from different sources might depend on each other. In particular, it uses a set of piece–wise constant dependency functions to capture the correlation between labels (i.e., data from different sources). It uses these functions to create a decision learning tree to describe the inter–dependency model. Based on this model, it then estimates the probability of event occurrence in the future by using forward and importance sampling (Gunawardana et al., 2011).
- **Dirichlet Process based (DP)**: this algorithm is designed for predicting the presence at locations of a single appliance ((Gao et al., 2012), (Tominaga et al., 2012)). In particular, it regards a set of binary observation (i.e., whether the user is at a certain place at a particular time), modelled by beta distributions. The parameters of these distributions are conditioned on a *day type* category for that particular day. To adapt this algorithm to our settings, we run it on each appliance, as if it was independent from the others.
- **An extension of the Dirichlet Process based (DP–Ext)**: this extends the model of Tominaga et al. (2012) to capture inter–dependency by assuming that if two appliances share the same day type class, they are highly correlated. In general, DP–Ext is capably dealing with the inter-dependencies between multiple appliances. However, it does not take into account the cyclic human routine.
- **EGH**: this is a rule–based method to find patterns of appliance usage events on the historical given preceding window (Laxman et al., 2008).

We refer to our algorithms as GM–MAP, and EGH-H. In the next section, we provide evaluation metrics for our experiments.

6.3 Evaluation Metrics

We evaluate the prediction accuracy of our algorithms against other benchmarks using the receiver operating characteristic (ROC), which illustrates the performance of the algorithms as the threshold of the algorithm is varied.[3] The ROC curve shows the trade–off between the algorithm's true positive rate (TPR) and false positive rate (FTR), which are defined as follows:

[3] The threshold value is varied from 0.0 to 1.0. Each threshold value can be used to estimate the True Positive Rate (TPR) and the False Positive Rate (FPR).

$$TPR = \frac{TP}{TP + FN} \tag{6.1}$$

$$FPR = \frac{FP}{FP + TN} \tag{6.2}$$

where:

- TP (True Positive) - the number of appliance usage that are predicted correctly as being used.
- FP (False Positive) - the number of appliance usage that are predicted as being used, but actually not being used.
- TN (True Negative) - the number of appliance usage that are predicted correctly as not being use.
- FN (False Negative) - the number of appliance usage that are predicted as not being used, but actually being used.

Given the ROC curve, we use the area under the ROC curve (AUC) to evaluate the prediction accuracy. The AUC is varied from 0.0 to 1.0 that represents from the worst prediction accuracy to the best prediction accuracy. We next turn to evaluate our algorithms, on the synthetic data in the following section.

6.4 Synthetic Data Evaluation

In this section, we evaluate the prediction accuracy on all the algorithms using a synthetic dataset. More importantly, using the synthetic dataset allows us to test the algorithms with different settings of inter-dependencies between appliances. Here, we basically concentrate on testing the prediction performance with a variety of level of the inter-dependencies for two appliances: washing machine and tumble dryer.

We assume that these appliances (i.e., washing machine and tumble dryer) are temporally dependent on each other. In particular, the use of washing machine implies the use of the tumble dryer. To do so, we assume that the tumble dryer is always turned on once the washing machine has been used (i.e., turned on and off). Here, the levels of inter-dependencies between these appliances are in relation to the gap between these two usage, and the frequency that the pair of these usage (i.e., same gap) can repeat. For example, considering a scenario that a user uses the washing machine every week, there are two cases of using the tumble dryer: i) a user will turn on the tumble dryer randomly during next 24 hours (equivalent to one day), ii) a user will turn on the tumble dryer randomly during next 169 hours (equivalent to one week). Intuitively, we can find that the relationship between these two appliances in the first case is stronger than the second case because it is more likely to find the repetitive pair (same gap) in the first case than the second case. Hence, we model the gap between these two labels by defining a dependency function $\digamma(\varpi)$ which randomly withdraws a number in a range of $1 \ldots \varpi$, where $\varpi \in \mathbb{Z}^{+}$ is the noise of the inter-dependency between the appliance usage. In this regard, assume the washing machine is being used every day, once the washing machine is being used, then the time that the tumble dryer is being used is defined as the following equation:

$$t_{td} = t_{wm} + \digamma(\varpi) \tag{6.3}$$

where t_{wm} is the time that the washing machine is being used that is randomly generated during the day. In this equation, if ϖ increases, the level of inter-dependencies decreases.

We generate two set of synthetic datasets:

- Synthetic Dataset 1 (SD1): we fix a number of usage of washing machine per day (arbitrarily set to 3), then we vary the level of dependencies based on varying the noise of the inter-dependency ϖ. Here, we want to evaluate the prediction performance on different level of inter-dependencies between appliances. We expect that the prediction accuracy would be likely to reduce if ϖ increases (i.e., the level of inter-dependency between appliances increases)
- Synthetic Dataset 2 (SD2): we fix the level of dependencies through ϖ (arbitrarily set $\varpi = 10$), then we vary the number of usage of the washing machine for a day. Here, we want to evaluate the prediction performance on the different number of usage of washing machine for a day. If these usage increases, there will be likely to find stronger dependencies between the two appliances. Thus, we expect that the prediction accuracy would be likely to increase if the number of appliance usage of the washing machine for a day increases.

These datasets are divided into two parts: training dataset, and testing dataset. The training dataset is generated from day 1 to day 30. The testing dataset is generated from data 31 to day 40.

Figure 6.1 shows the prediction performance of all the algorithms (including benchmarks) on SD1. In overall, the prediction accuracy (represented by AUC) for all algorithms decreases when the noise of the inter-dependency increases (equivalent to the decrease of the level of inter-dependency between the appliances). GM-PMA has shown to outperform all other algorithms. In particular, GM-PMA outperforms EGH-H (the second best performance) as the AUC of GM-PMA on average is approximately 5% higher than EGH-H. In addition, Figure 6.2 shows the prediction performance of the

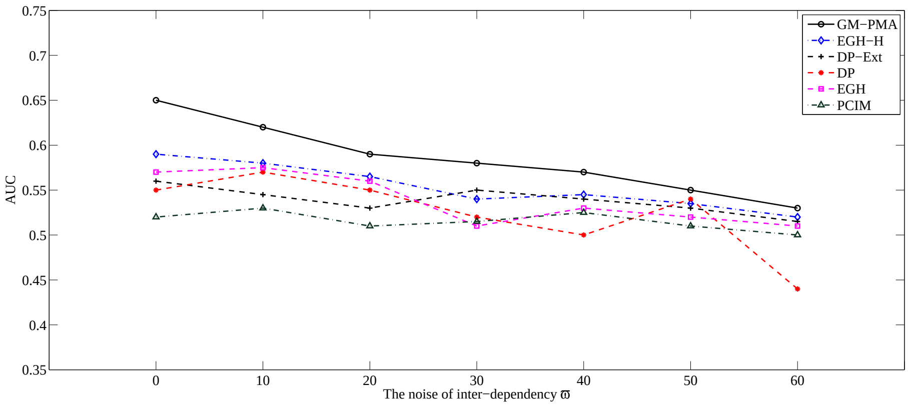

Figure 6.1: Synthetic data evaluation for two appliances: washing machine and tumble dryer. The number of usage of washing machine is arbitrarily fixed at 3. The noise of inter-dependency between the appliances is varied. Note that the higher the noise of inter-dependency value are generated, the lower the level of inter-dependency between the two appliances is likely to achieve.

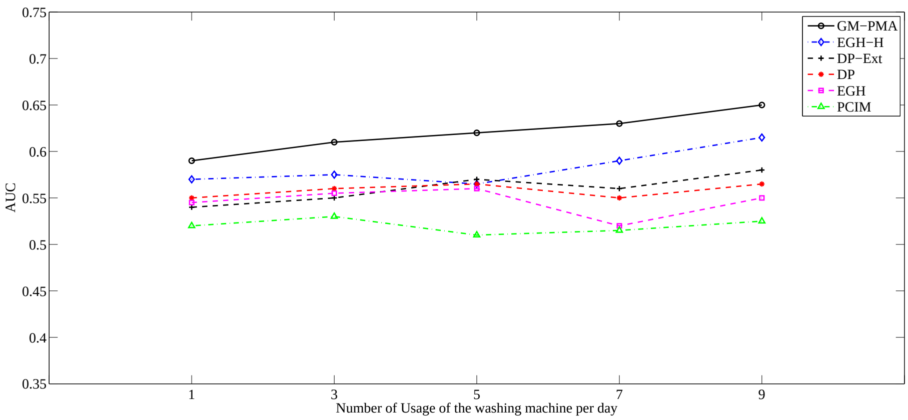

Figure 6.2: Synthetic data evaluation for two appliances: washing machine and tumble dryer. The level of inter-dependency between these appliances is fixed.

algorithms on SD2. The prediction accuracy of all algorithms is likely to improve if the number of usage of the washing machine for a day increases. Similarly, GM-PMA also dominates other algorithms, and the AUC of GM-PMA outperforms EGH-H (the second best performance) approximately 6% higher.

In the next section, we empirically evaluate the prediction performance on real–world datasets.

6.5 Empirical Evaluation on Real–World Datasets

In this section, we analyse the results of all algorithms in the two real–world datasets: the REDD dataset, and the FigureEnergy dataset. In particular, we first evaluate the

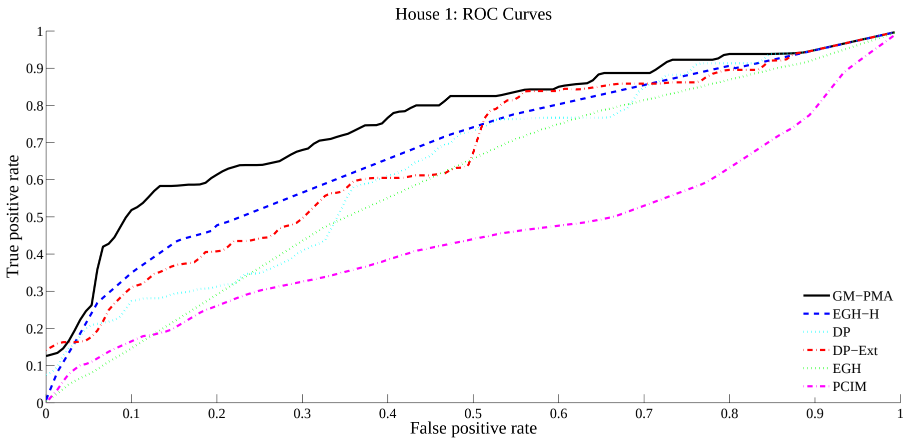

Figure 6.3: ROC curve of the algorithms run on house 1 - REDD.

prediction accuracy on the REDD dataset in Section 6.5.1. Similarly, we show the results on the FigureEnergy dataset in Section 6.5.2. We then evaluates human routine cycles in Section 6.5.3. Finally, we analyse the algorithm's runtime in Section 6.5.4.

6.5.1 Experiment 1: Performance on the REDD Dataset

Here, we run our algorithms to predict all the labels of the REDD dataset. We depict the Receiver Operating Characteristic (ROC) curve of the algorithms for each home in Figure 6.3, Figure 6.4, and Figure 6.5. From Figure 6.3, we can see that our algorithm dominates all the others. In particular, the area under the curve (AUC) of GM–MAP in home 3 is 0.79, while the AUC value for DP, DP–Ext, PCIM, EGH-H, and EGH is 0.58, 0.65, 0.38, 0.67, and 0.58 respectively. In other words, our algorithm (GM–MAP) outperforms DP, DP–Ext, PCIM, EGH-H, and EGH by 21%, 14%, 41%, 12%, and 21% respectively in home 1. Similarly, our algorithm dominates the nearest best algorithm DP-Ext up to 15% in home 3. Note that since home 1 and home 3 have the most detailed data,[4] all the algorithms typically exhibit their best performance on these homes. Note that the data from home 4 is less detailed, and thus, all the algorithms perform worse, compared to themselves in home 3 and home 1. However, our algorithm still dominates the benchmarks. An exception is the PCIM method, which performs by far the worst. The reason for this is that due to the large size of the dataset, PCIM overfits the inter–dependency model (since it does not take into account the cyclic feature of human routine). Given this, it fails to correctly detect the occurrence of activities.

6.5.2 Experiment 2: Performance on the FigureEnergy Dataset

In this section, we test the performance of our algorithms on two selected homes from the FE dataset as the other homes did not have sufficient data for the algorithms to

[4]The number of labels in home 1 and 3 is higher than the number of labels in home 4.

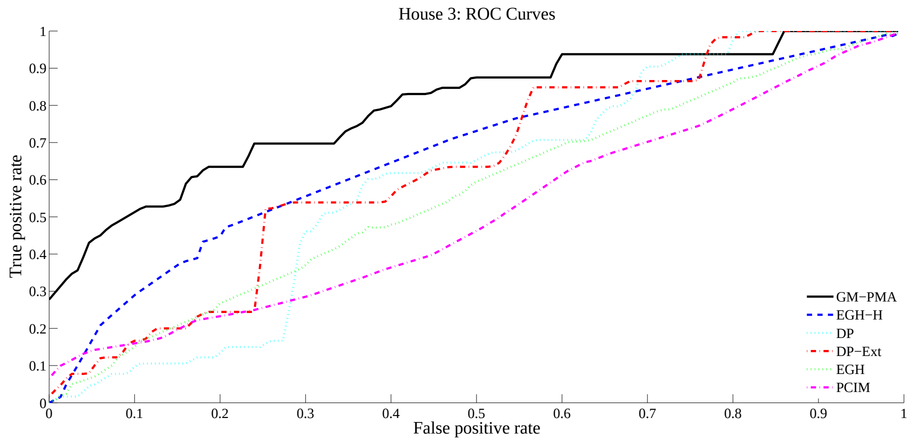

Figure 6.4: ROC curve of the algorithms run on house 3 - REDD.

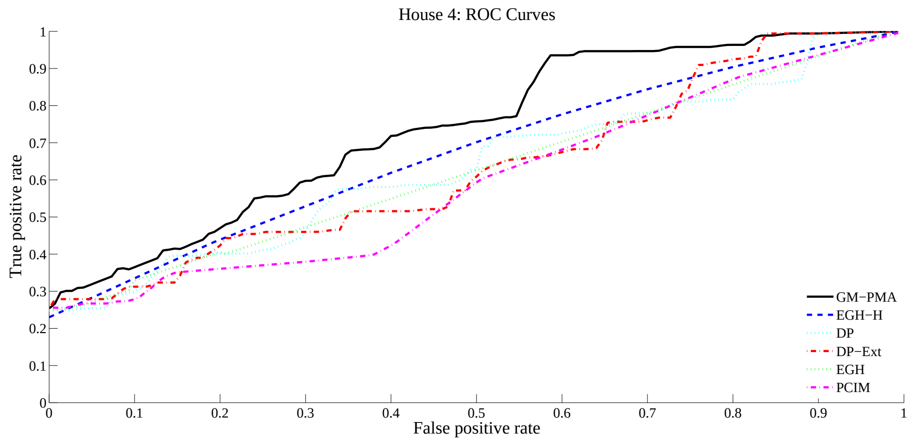

Figure 6.5: ROC curve of the algorithms run on house 4 - REDD.

be trained and tested. Thus, we were not able to set up a proper training dataset for those homes. Similar to the previous section, we also consider of measure the prediction accuracy of the algorithms by using the ROC curves measurement. Note that in the FE dataset, the labels of energy usage activities were annotated by consumers, and hence, there is a possibility that users might mistakenly select incorrect activity types or durations of specific activities. Thus, the uncertainty of the labels is high. Consequently, the algorithms can be overfitted in training, and hence worsen the prediction performance.[5] Therefore we selected labels that occurred sufficiently in both training and test datasets. We also plot the ROC curve of the algorithms for these homes in Figure 6.6 and Figure 6.7. From these figures, we can observe that, due to the uncertainty of the users' manual labelling process, the performance of the algorithms is much worse, compared to the case of the REDD dataset. However, GM–MAP still provides the highest accuracy in predicting future activities. For example, GM–MAP outperforms DP, DP–Ext, EGH-H, EGH and PCIM by approximately 12%, 10%, 11%, 20% and 30% respectively.

[5]This is an aspect which we will further investigate as part of our future work.

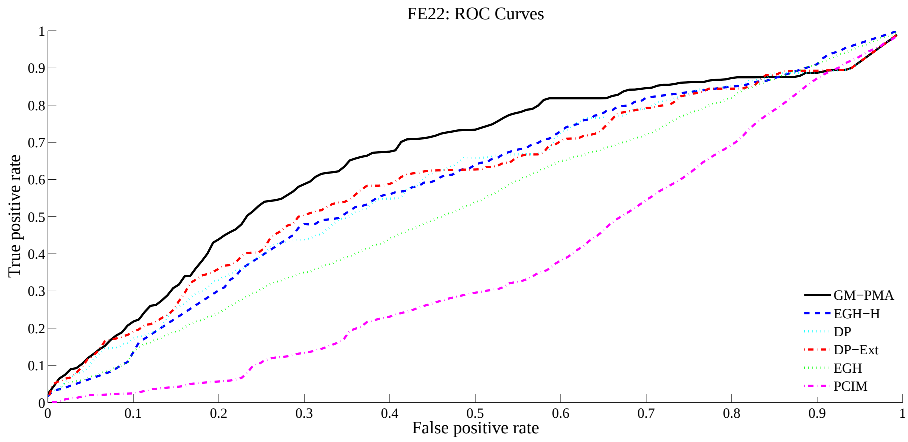

Figure 6.6: ROC curve of the algorithms run on User FE22 from FigureEnergy.

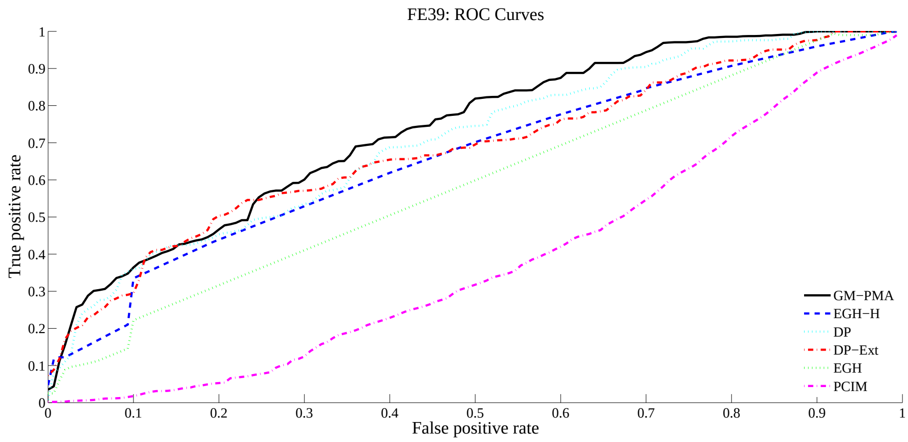

Figure 6.7: ROC curve of the algorithms run on User FE39 from FigureEnergy.

6.5.3 Experiment 3: Human Routine Cycles

In this experiment, we show that our appliance usage prediction algorithms (GM-PMA and EGH-H) will improve the prediction accuracy if we take into account weekly periodicities of human routine behaviour. In particular, we already demonstrated that GM-PMA outperformed DP-Ext using real-world datasets in Experiment 1 and 2. Note that GM-PMA learns day types classes conditioning on the day of the week, while DP-Ext does not.

Moreover, with regards to EGH-H, to test the appliance usage prediction with different periodicities, we use the REDD dataset (house 1).[6] Figure 6.8 shows the performance of usage prediction with different cycle routines (ranging from 1 to 7 days) for the EGH-H algorithm. We can see that the prediction accuracy of EGH-H with a weekly period is much improved compared to EGH-H with other periods. In more detail, EGH-H with

[6]House 1 in the REDD dataset provides more appliances than others. We only use this to strengthen the use of weekly human routine assumption in our model.

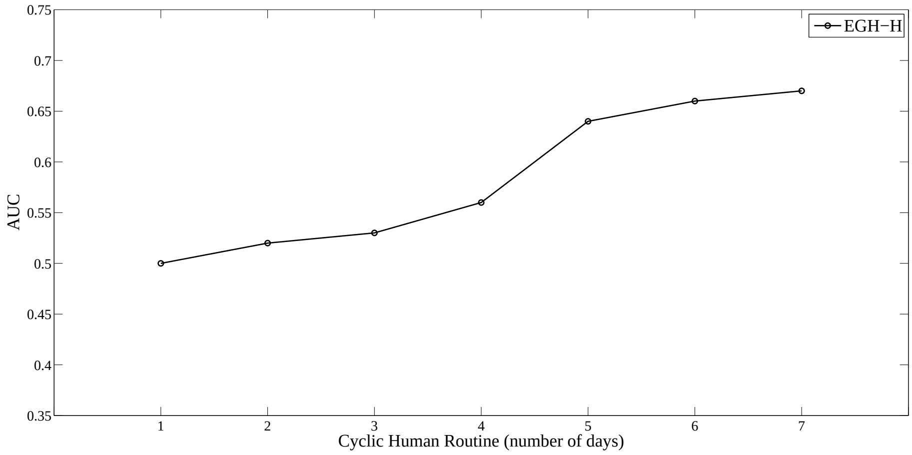

Figure 6.8: Cyclic human routine evaluation on EGH-H for REDD (House 1) dataset.

GM-PMA	EGH–H	DP	DP-Ext	PCIM	EGH
545.86	2.58	1225.14	815.24	371.30	1221.72

Table 6.1: Runtime performance for 8 appliances (in seconds).

a weekly period dominates EGH-H with one day period approximately 17% in term of the AUC measurements.

6.5.4 Experiment 4: Average Runtime

Having evaluated the prediction accuracy of the algorithms, we now turn to evaluate the running time of each algorithm. In particular, we run the algorithms on an Intel(R) Xeon(R) computer (64–bit operating system) with 2.67 GHz and 12GB. The results measured in seconds are depicted in Table 6.1. We measure the runtime of each algorithm on the REDD dataset (house 1) within 8 appliances. We can see that even in the case of 8 labels, the runtime of the benchmark algorithms is significant (9, 20, 14, 6 and 21 minutes for GM-PMA, DP, DP-Ext, PCIM, and EGH). In contrast, the runtime of EGH–H still remains under 3 seconds. This implies that EGH–H could be you for scenarios that feedback may need to generate in short notice. Additionally, EGH-H could be used for interactive feedback, where the agent suggests users different home energy consumption plans in real–time (if needed), as it can use EGH–H to quickly predict the next–day usage, based on the real–time feedback of users. On the other hand, GM-PMA may provide more accurate schedule of appliance usage, but this algorithm requires more time to generate. Thus, it is best use for scenarios that users are comfortable of waiting. For example, the agent can request GM-PMA algorithm to run at the end of the day, then provide users the appliance usage plan in the morning. We now summarise the findings of this chapter in the following section.

6.6 Summary

In this chapter, we provided the experimental settings for our evaluation on the real–world datasets. We described a set of benchmark algorithms that we used to compare to our algorithms, including PCIM, DP, DP-Ext and EGH (these algorithms have also described in Chapter 2). We also provided evaluation metrics such as ROC curves and AUC to measure the prediction performances for the prediction algorithms. We then simulated the synthetic dataset with two appliances (a washing machine and a tumble dryer) to evaluate the prediction performance of all the algorithms. The use of synthetic data allows us to test with different settings of inter-dependencies between these two appliances. We found that GM-PMA dominated other algorithms using the synthetic datasets. Then, we demonstrated the prediction performance of our algorithms with other benchmark algorithms using the two real-world datasets: the REDD dataset and the FE dataset. We found that GM-PMA algorithm outperformed existing methods by up to 41% in term of prediction accuracy. Similarly, EGH-H dominates other benchmarks by up to 420% faster in term of runtime performance.

So far, we have addressed the appliance usage prediction problem by proposing efficient algorithms: GM-PMA, and EGH-H. In the next chapter, we show how to use our prediction algorithms, and provide insights about how the agent can assist to minimise the user's discomfort and costs as a result of deferring appliance usage.

Chapter 7

Intelligent Demand Response

In this chapter, we study a scenario in which an agent assists to optimise their savings and comfort using GM-PMA. In particular, we propose an Intelligent Demand Response (IDR) mechanism that extends GM-PMA to learn, from the user's responses, the best suggestions to give to the user to maximise the acceptance of the load deferment. The IDR mechanism has three main phases, include: i) Appliance usage prediction (Phase 1), ii) Suggestion optimisation (Phase 2), and iii) Human-Agent interaction (Phase 3). We describe these phases more formally in Section 7.1. In order to evaluate IDR (particularly for Phase 3), we model and simulate realistic human responses to suggestions in Section 7.2. In Section 7.3, we elaborate on Phase 2 to show how the agent can optimise *personalised* suggestions. Then, we describe how the agent can learn the user's preferences (i.e., the trade–off between the user's comfort and savings) from the user's responses (i.e., Phase 3) in Section 7.4. Given this, we empirically evaluate the IDR mechanism using the REDD dataset in Section 7.5. Finally, we summarise this chapter in Section 7.6.

7.1 The IDR Mechanism

The IDR mechanism iteratively runs through the three main following phases:

- **Appliance usage prediction** (Phase 1): here, we wish to predict which appliances are likely to be used, and when they are likely to be used during the day. This prediction problem has been analysed and addressed in Section 3. In particular, within the home, there is a finite set of appliance usage events, where different types of appliances are distinguished by labels $l \in L$, and the time of the day $t \in T$ can be discretised by a number of time slots for a day (e.g., T = 24 is equivalent to 1 hour time slot). $x_{n,l,t}$ indicates whether appliance l was used on day n at time t.

Here, we employ the GM-PMA algorithm to solve this problem given its superior performance in terms of accuracy.[1]

- **Suggestion optimisation** (Phase 2): by considering the predicted activities of the user (based on Phase 1) as well as the electricity prices (here we assume a dynamic pricing regime (Ramchurn et al., 2011a)), the agent can optimise a suggestion plan of the use of appliances that will be passed to the user to maximise their savings and comfort. In particular, we denote $p_n^t \in R$ as the electricity price at time slot t on day n. According to the electricity price, then the agent estimates optimal suggestions that can help the user save money. A suggestion is in the form of a tuple $< S(l)_n^{t_i \to t_j}, \Delta t, \Delta E(l)_n^{t_i \to t_j} >$, where $S(l)_n^{t_i \to t_j}$ indicates an appliance l on day n should be deferred from time slot t_i to time slot t_j, $\Delta t = |t_i - t_j|$, and $\Delta E(l)_n^{t_i \to t_j} \in R^+$ is potential savings gained if the user would accept this suggestion. We describe this phase in more detail in Section 7.3.

- **Human-Agent interaction** (Phase 3): the agent chooses the best *personalised* suggestions, and sends them to the user. For example, the agent may suggest the deferment of the washing machine from 8.00am to 10.00pm because it is cheaper to do so. Then, the user responds to the agent's suggestion by agreeing or rejecting the proposal. Formally, a user's response given to the suggestion $S(l)_n^{t_i \to t_j}$ is noted as $D(l)_n^{t_i \to t_j} = (0, 1)$, where 0 means the user does not follow the agent's suggestion, and 1 means the user would follow the agent's suggestion. Crucially, by analysing many such responses over time, the agent can learn the user's preferences from her reactions in order to select better suggestions in Phase 2. The details of this phase are described in Section 7.4.

Now, in order to evaluate IDR, we need to model and simulate realistic human responses to suggestions (in Phase 3). Hence, we describe human response model in the next section.

7.2 The Human Response Model

Human decision–making (as it relates to appliance usage) is clearly complex and determined by many factors. Also, there is not enough real data to understand user's behaviour in this domain. Thus, completely modelling the user's behaviour would be unrealistic. However, many studies have shown that the energy consumption behaviours are mainly influenced by cost, comfort and even peer-pressure[2] (Froehlich et al., 2010). Given this, we use these factors (except peer-pressure as it does not apply to our settings)

[1] Since the focus here is on learning from human responses rather than providing timely feedback, we only use GM-PMA.

[2] More information can be seen at http://opower.com/solutions/behavioral-demand-response, but testing on peer-pressure is beyond the scope of this paper.

to model human response in our scenario. Note that more complex human behaviour model will be considered as our future work. In more detail, we first describe a user's utility function that defines the agent's interpretation of the user's preferences in Section 7.2.1. Then, we elaborate on how we develop a decision–making model given agent suggestions in Section 7.2.2.

7.2.1 User Utility Function

In the context of the home energy management, energy consumption reduction or deferment results in monetary savings (i.e., as per the user's energy tariff). However, people may have their own preferences to use specific appliances at certain times during a day (e.g., prefer cooking at 7.30pm, wash and dry their clothes during the weekend). In addition, the use of appliances in the home is usually constrained by daily activities (e.g., working hours in offices, spending time for kids, weather forecast). Thus, there exists inconvenience/loss of comfort associated with using appliance at time t_j instead of t_i.

Hence, savings may be in tension with comfort (i.e., there is a trade–off). To satisfy a user's preferences and (possibly) her real-life constraints (*via* their appliance usage), the agent can make suggestions to change the user's existing appliance usage behaviour by accounting for the trade-off between comfort and savings. In particular, to represent this trade–off, we let utility of user following agent suggestion be f_n: $U(s, t_i, t_j)$, where s is the monetary savings to be made for accepting the agent's suggestion to defer appliance usage from time t_i to t_j. Here, t_i is the preferred time that the user would like to turn on the home appliance. Any suggestion that asks the user to move away from this time is likely to cause a loss of comfort. Thus, the assumption we make for estimating the user's discomfort in our model is that the more the user is asked to deviate from the time they would prefer to turn the the appliance on, the more discomfort it will cause to the user. As such, the discomfort function is assumed to be stationary (e.g., that the discomfort of going from 2pm to 4pm is identical to the discomfort of shifting from 2pm to 12 noon or from 4pm to 6pm).[3] Moreover, the monetary savings s is the difference between the energy cost at time slot t_j and the energy cost at time slot t_i, which can be estimated as follows:

$$s = \Delta E(l)_n^{t_i \to t_j} \quad = \quad E(l)_n^{t_j} - E(l)_n^{t_i}$$

where $E(l)_n^{t_j}$ is the energy cost of appliance l at time slot t_j, and $E(l)_n^{t_i}$ is the energy cost of appliance l at time slot t_i. The energy cost of an appliance l at time slot t on day n can be estimated as follows:

[3]More complex models are beyond the scope of this paper and will be considered in future work.

$$E(l)_n^t = d_l c_l p_n^t \tag{7.1}$$

where d_l (hours) is the duration of operation for appliance l, c_l (kW) is the power consumption of appliance l, and p_n^t (£/kWh) is the energy cost at time t on day n. In our experiments, we use real data for the average duration of the appliance and the energy consumption of the appliance that have been measured and collected in some sources.[4] For example, a dishwasher typically requires approximately 1.85 (kWh) in 2 hours of operation, while a washing machine consumes around 0.63 (kWh) in 1.5 hours on average.

To model the user's utility function, for the sake of simplicity, we use a standard linear model as used in many scenarios (Vytelingum et al., 2010; Ramchurn et al., 2011a). More specifically, the user's utility function can be written as:

$$U(s, t_i, t_j) = U(l)_n^{t_i \to t_j} = w_c|t_j - t_i| + w_s s$$

where $w_c \in \mathbb{R}$ is the weight of the comfort of the user and $w_s \in \mathbb{R}$ is the weight of the monetary savings of the user. The unit of w_c is $£/(\text{time slot})$. As we will arbitrarily fix $w_s = 1$ in our experiment, thus we only need to focus on w_c (see details in Section 7.5.1).

So far, our model specifies a user's preferences given suggestions but does not specify how we simulate the user's responses to the agent's suggestions. We address this in the next section.

7.2.2 Decision–making given suggestions

In an ideal scenario, the user would say 'yes' with probability 1 to any suggestion that had positive total utility for them, and 'yes' with probability 0 to any suggestion that had negative total utility (i.e., 'no'). This is equivalent to a step function applied to the total utility. However, even if we assume our model of user responses is completely correct (i.e., that their decision boundary between comfort and savings is linear), people do not always behave rationally very near to their decision boundary (i.e., occasionally, they may say 'no' to perfectly good suggestions) (Costanza et al., 2014). To improve the realism of our model of user responses, we use logistic regression as our decision–making model for users to deal with any given suggestions. More practically, it is more convenient to perform logistic inference with differentiable functions (which the step function is not), and the sigmoid function is by far the most widely used function for this purpose in settings that have binary classifications (Bishop, 2006). Formally speaking, we apply a sigmoid function within the user utility function (Equation 7.2) to generate user responses to suggestions ('yes' to accept and 'no' to reject). Then, the

[4] http://www.daftlogic.com/information-appliance-power-consumption.htm.

likelihood of the user to accept a suggestion can be estimated as a sigmoid function as follows:

$$\begin{aligned} p(D(l)_n^{t_i \to t_j} = 1|\phi) &= \sigma(w^T \phi) \\ &= \frac{1}{1 + \exp(-U(s, t_i, t_j))} \\ &= \frac{1}{1 + \exp(-U(l)_n^{t_i \to t_j})} \\ &= \frac{1}{1 + \exp(-(w_c(\Delta t) + w_s(\Delta E(l)_n^{t_i \to t_j})))} \end{aligned}$$

Using this sigmoid function, it is thus possible to simulate the user responses to the agent's suggestions. As this model is dependent on the two parameters w_c and w_s, some settings of these parameters would give identical behaviour in our model. We address this in Section 7.5.1. Now, we turn to describe how the agent optimises the user's personalised suggestions (Phase 2) in the next section.

7.3 Suggestion Optimisation

In this section, we describe an optimisation algorithm for selecting personalised suggestions that can maximise user's comfort and savings. At this stage, the agent already knows the user's appliance usage plan *via* solving the appliance usage prediction problem (in Phase 1). Taking into account the forecast of the electricity price, the agent solves the user's suggestion optimisation problem by maximising the user's utility function (see Equation 7.2). In more detail, we denote $\chi(l)$ (initially empty) as a set of appliance usages l to be deferred (obtained from prediction appliance usage in phase 1), then the optimisation process can be done as follows. First, we select the time to be deferred that has optimal utility as:

$$\begin{aligned} S(l)_n^{t_i \to t_j} &= arg \max_{t_i, t_j \in T} U(l)_n^{t_i \to t_j} \\ s.t. &\quad t_j \notin \chi(l) \ \& \ t_i \in \chi(l) \end{aligned}$$

Second, we update time to be deferred t_j for appliance l.[5]

$$\chi(l) = \chi(l) \cup t_j \tag{7.2}$$

According to the calculation from the optimisation process, the agent then can choose the best *personalised* suggestions for the user. We send these suggestions to the users to

[5]Please note we only deal with binary decision for this scenario. More complex decision will be considered as future work.

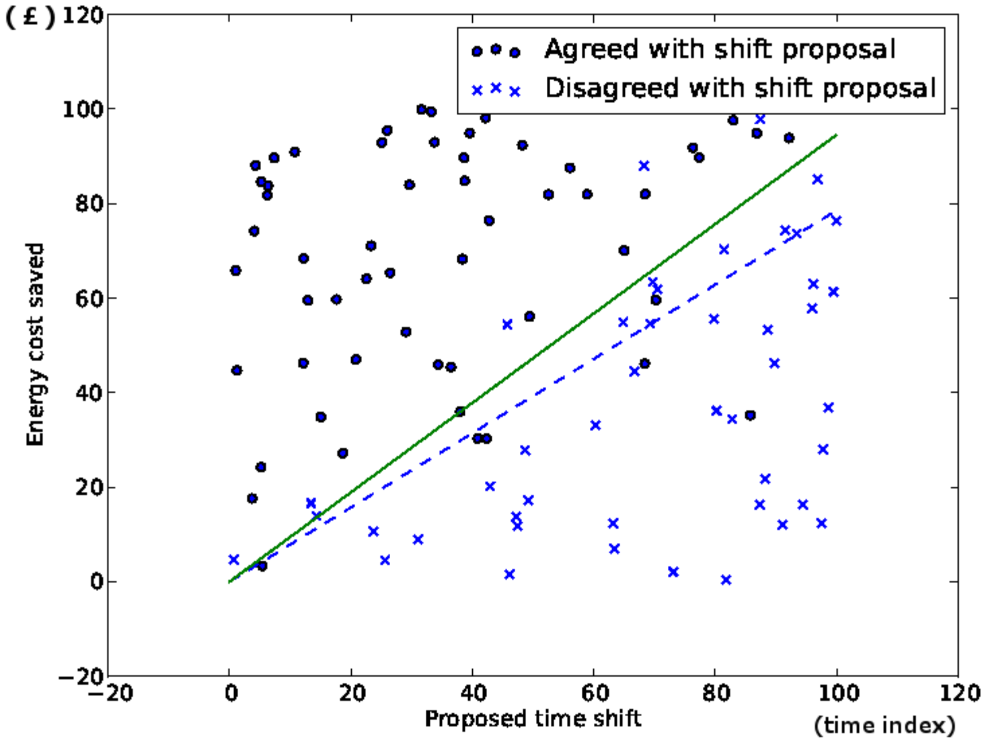

Figure 7.1: Example of using logistic inference to learn the user's preferences from the user's responses. The algorithm finds a new gradient (the dashed line) to minimise the errors of agent's suggestions. Intuitively, the user would say 'yes' to the suggestion that the trade-off located above the line, and 'no' otherwise.

obtain their responses. In the next section, we describe how the agent learn the user's preferences from the user's responses to the agent's suggestions (Phase 3).

7.4 Learning from User Responses

Here, we model the interactions between the user and the agent and develop an algorithm that can adapt to the user's changing preferences by learning over the user's responses to the agent's suggestions. This learning process can be summarised as follows. First, by maximising the utility function, the agent selects a list of *personally meaningful* suggestions $s_{1..M} = \{s_1, \ldots, s_M\}$, where $s_i = S(l)_n^{t_i \to t_j}$ (see Phase 2). For each suggestion s_i, the agent has already estimated the trade–off between comfort and savings, noted in the form of vector $\boldsymbol{\phi_i} = (\Delta t, \Delta E(l)_n^{t_i \to t_j})^T$. Second, by sending these suggestions to the user, the agent obtains a list of corresponding responses from the user $d_{1..M} = \{d_1, \ldots, d_M\}$, where $d_i = (0, 1)$. Here, $d_i = 0$ means the user says 'no' to the suggestion, and $d_i = 1$ means the user says 'yes' to the suggestion. Third, we learn the user's preferences by adjusting the weights (of discomfort and savings) for the user's utility function (see Equation 7.2). Note that we select the best *personalised* suggestions based on the user's utility function in Phase 2. More importantly, we learn these weights by using logistic inference (Bishop, 2006). Given the agent's suggestion trade-offs $\boldsymbol{\phi_{1..M}}$, and the corresponding user's responses $d_{1..M}$, the logistic inference algorithm returns the corresponding learnt weights for comfort (w_c) and savings (w_s) that can minimise the error of the agent's suggestions. These weights will be used for the next iteration of our IDR mechanism (Section 7.1). Figure 7.1 illustrates how the logistic inference algorithm works in our scenario. Next, we empirically evaluate our IDR mechanism to determine the potential money that can be saved for the user.

7.5 Empirical Evaluation

We evaluate the IDR mechanism on the REDD dataset with a number of simulated human responses in two experiments. In Experiment 1, we analyse whether users are inclined to make more savings by using the IDR mechanism. We also evaluate the logistic inference algorithm that learns user's preferences. We describe this experiment in more detail in Section 7.5.2. In Experiment 2, we evaluate the process of selecting the best *personalised* suggestions in Phase 2 (Section 7.5.3).

In addition, to evaluate IDR under reasonable real-time pricing schemes, we simulate real-time prices as Ramchurn et al. (2011a).[6] In turn, given the lack of real data about user's response behaviours (due to the novelty of proposed system and unknown ground truth that would help us determine the solution), we use synthetic data to model different human response behaviours. More importantly, the use of synthetic data not only helps us to consider broad range of behaviours, but also allows us to study how quickly inference gets close to the true value. We describe how we generate user profiles for our experiment in the following section.

7.5.1 User Profiles

As described in Section 7.2, our human response model has two parameters, w_s and w_c, representing the user's personal preference for energy savings and comfort, respectively. In our experiments, we want to test the improvement in performance that our system brings with respect to a variety of simulated human response behaviours. To create such behaviours, we must vary the ground truth savings and comfort parameters. However, we notice that some settings of w_s and w_c would give identical behaviour in our model *before* response noise is added (i.e., before applying the sigmoid function). This is due to the linearity at the utility function (see Figure 7.2). In order to consider a wider variety of behaviours, we therefore avoid considering ground truth parameters w_s and w_c that correspond to identical noiseless behaviours. This can be most conveniently achieved by

[6] The approach generates the prices from simulated aggregate demand on the grid.

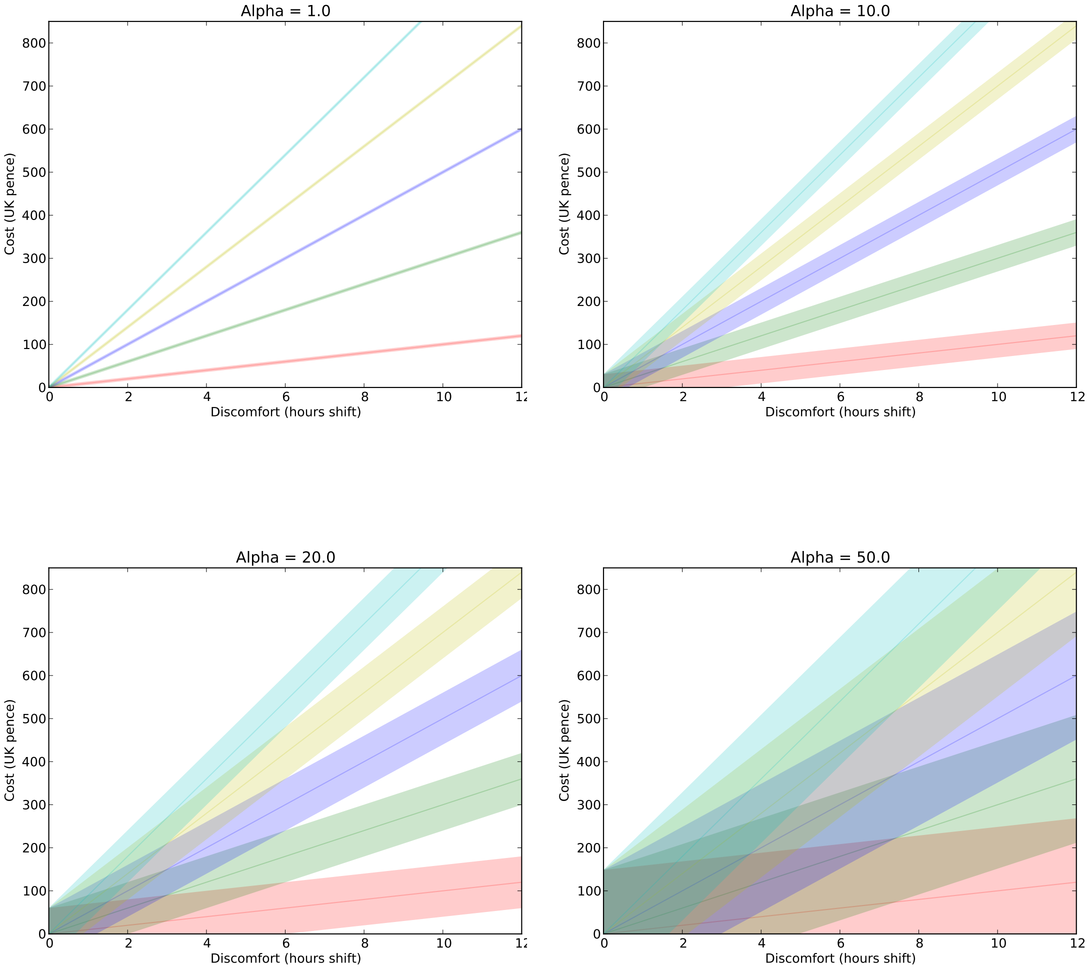

Figure 7.2: The 5 canonical ground truth user preferences we use in our experiments, represented as lines of different colour (representing the linear trade-off between savings and comfort). For savings-comfort values *above* each line, the rational response (i.e., without noise) for the simulated user is to accept the proposed appliance usage shift. Otherwise, the suggestion is rejected. However, since our model is probabilistic, the user may behave irrationally. The extent of irrationality is best understood as new parameter α under a change of basis for the utility function $U(s, t_i, t_j) = \frac{1}{\alpha}(w_c|t_j - t_i| + s)$. The 4 subfigures show different values for irrationality. The shaded areas indicate the 95% confidence region that the simulated response behaviour will be irrational (i.e., be the opposite response to the one dictated purely by utility theory). In our experiments, we fix $\alpha = 1$ (equivalent to $w_s = 1$ in the original basis given in Equation 7.2) as a reasonable choice of irrationality, resulting in the need to change only one parameter (w_c) throughout.

UserID	0	1	2	3	4
w_c	-10	-30	-50	-70	-90

Table 7.1: Different User settings for different decision making.

fixing one of the parameters (we arbitrarily chose $w_s = 1$) while allowing the other (w_c) to vary. The various parameter settings we considered in our experiments are given in Table 7.1. We expect to see that the user with higher adaptability (i.e., higher value of w_c) will be likely to save more money. We use these user behaviours in Experiment 1 in the next section.

7.5.2 Experiment 1: Evaluating IDR

In this section, we evaluate the IDR mechanism under a number of simulated human response behaviours (that has been defined in Section 7.5.1) using the REDD dataset. Settings for the training set and the testing set for each home in the REDD dataset are set similar to the settings for appliance usage prediction problem. Thus, we apply IDR over these test dataset to test two hypotheses:

- **H1**: any user is likely to achieve more savings by using the IDR mechanism. Note that the IDR provides personalised suggestions to the user based on learning the user's preferences from the responses. Due to the uncertainty of suggestions, the user can reject suggestions and therefore impact the learning phase of the algorithm. Savings can only improve if suggestions are accepted by the users.
- **H2**: the user that tends to be more adaptive to suggestions (i.e., higher value of w_c) can save more money than the one that is conservative (i.e., avoids deferments). We show that the agent can computerise the user's behaviour by learning and suggesting the suitable trade–off between comfort and savings.

Our approach to test these hypotheses can be summarised as follows: First, we apply GM-PMA to predict all appliance usages for the next day. Second, we obtain all personalised suggestions from Phase 2. These suggestions are selected based on the learnt user preferences (i.e., maximising the user's utility function).[7] Third, the agent sends all personalised suggestions to the user and receives the corresponding responses. Then, we estimate the electricity cost at the suggestion's time if the user says 'yes' (i.e., accepts to defer). Otherwise, if the user says 'no' to the suggestion, the user will use the appliance at their preferred time (i.e., at the prediction time). Hence, we then calculate the electricity cost at the predicted time. The total potential savings for each user is the difference between the cost of the user's preferred time (i.e., at the predicted time)

[7]Note that, for this demonstration, we do not take into account the prediction error in Phase 1. We will consider it in the optimisation process in Phase 2 in the future work.

UserID	0	1	2	3	4
HomeID = 1	**1262.35**	1029.61	874.81	717.98	555.04
HomeID = 3	**2623.45**	2483.95	2119.33	1603.08	1374.06
HomeID = 4	**414.46**	276.83	253.29	236.24	217.46

Table 7.2: Savings (in UK pence) achieved by different simulated human response behaviours performed on REDD dataset.

and the cost of the user's deferred time. Table 7.2 shows the final results of how much savings each user can achieve. The results confirm that the users can always improve their savings if they use IDR (H1). Also, the most money can be saved by the user 0 who tends to adapt to the agent's suggestion, and the least money can be saved by the user 4 who tends to be more conservative with their preferred time for appliance usage (H2). In more detail, the user 0 gains approximately £12.62, £26.23, and £4.14 per week (i.e., the testing period) within House 1, 3 and 4 respectively, about 227%, 200%, and 100% respectively higher than the user 4.

To evaluate the logistic inference algorithm for learning the user's preferences, we estimate the error between the preset weight parameters w_s and w_c (these parameters are represented for the user's behaviour as defined in Section 7.5.1) and the new weight parameters $w_s^{'}$ and $w_c^{'}$ returned by the algorithm from learning the user's responses. In addition, as shown in Section 7.5.1, we have arbitrarily chosen $w_s = 1$. Therefore, we only need to evaluate the parameter w_c. To do so, we first normalise the weight of $w_c^{'}$. The normalisation equation is estimated as follow:

$$w_c^{norm} = \frac{w_c^{'}}{w_s^{'}} w_s$$

We define *learning error* is the difference between parameters w_c and w_c^{norm}. We measure the learning error to evaluate the performance of learning the user profiles. Intuitively, if this converges to 0, the logistic inference is efficient.[8] Figure 7.3, Figure 7.4, and Figure 7.5 show the learning errors that vary for house 1, 3, and 4 respectively. The value of the learning errors starts converging after the first 3 days. Also, the user with higher adaptive behaviours converges faster.

7.5.3 Experiment 2: Evaluating Personalised Suggestions

In this section, we evaluate the process of selecting the best *personalised* suggestions to the user. We consider the fact that the user will probably ignore too many suggestions. Thus, we select the best suggestion for each appliance to send to the user each day. More particularly, we want to test that the user can make the most savings if the agent

[8] We only use the learning error to validate the learning user behaviour. Other techniques can be used, but this is not a main focus here.

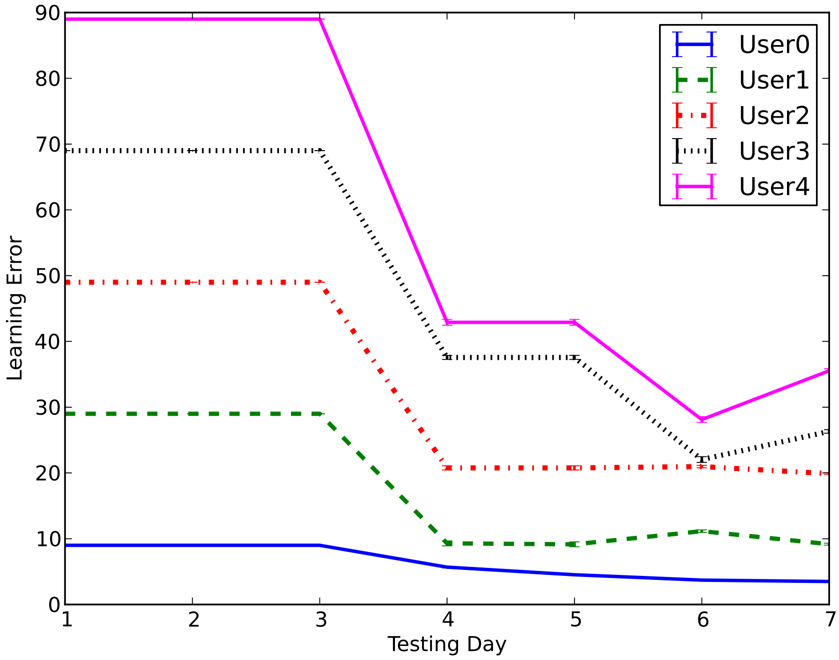

Figure 7.3: Learning user's behaviour error at House 1 (REDD dataset).

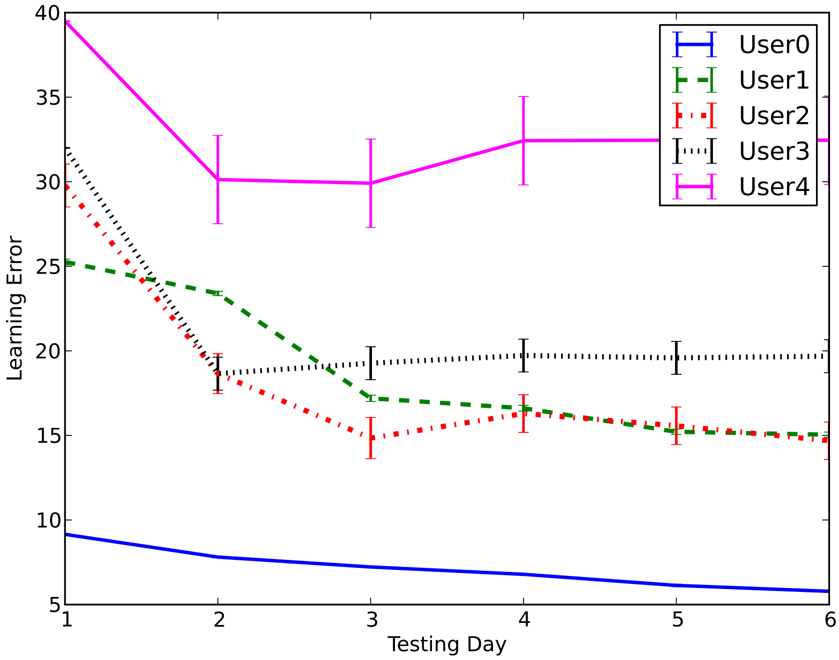

Figure 7.4: Learning user's behaviour error at House 3 (REDD dataset).

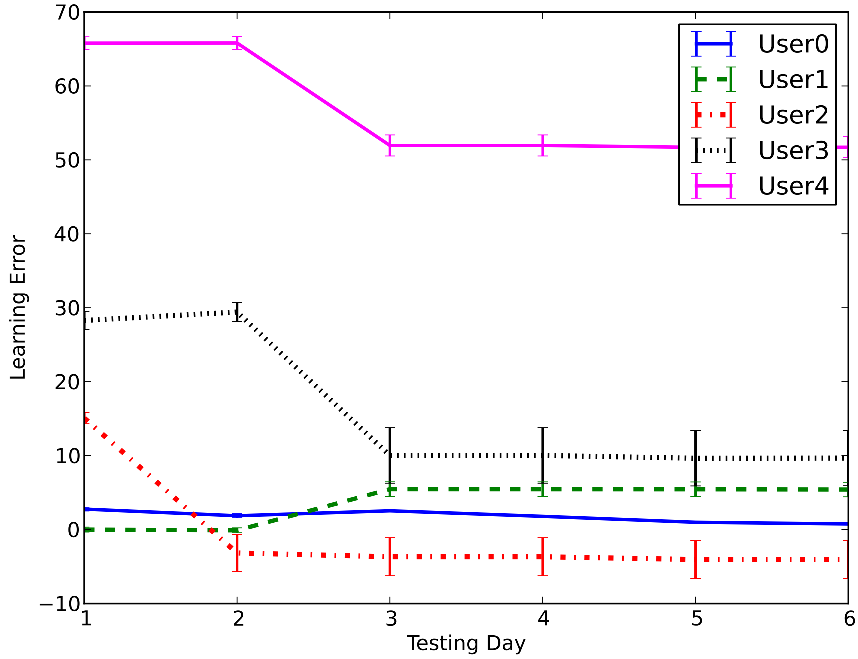

Figure 7.5: Learning user's behaviour error at House 4 (REDD dataset).

assists them with the best personalised suggestions that can satisfy her preferences (i.e., maximising the user's utility function in Phase 2). We call the algorithm that solves Equation 7.2 $AlgU$. We benchmark $AlgU$ with other two suggestion optimisation methods as follows:

- $AlgUs$: selects the best suggestion per appliance per day that returns the optimal user's savings (i.e., maximum $w_s s$ in the user's utility function).
- $AlgRand$: selects the best suggestion per appliance per day randomly.

Figure 7.6, Figure 7.7, and Figure 7.8 show the potential savings that the users can gain based on responding to the agent's *personalised* suggestions within House 1, 3, and 4 respectively. As we can see from this graph, $AlgU$ (that maximises the user's utility function) outperforms others with respect to savings (for a week of testing period). In particular, the $AlgU$ algorithm can save up to about £1.00/week (equivalent to £52.00/year) for user 0, and about £0.50/week (equivallent to £26.00/year) for user 4 within House 1. Similarly, $AlgU$ can save up to £1.20/week (equivalent to £62.40/year), and about £0.90/week (equivalent to £46.80/year) for user 0, and 4 respectively within House 3. In average, users using $AlgU$ can save up to 40%, 90%, and 70% in House 1, 3 and 4 respectively.

Now, we summarise the findings of this chapter in the next section.

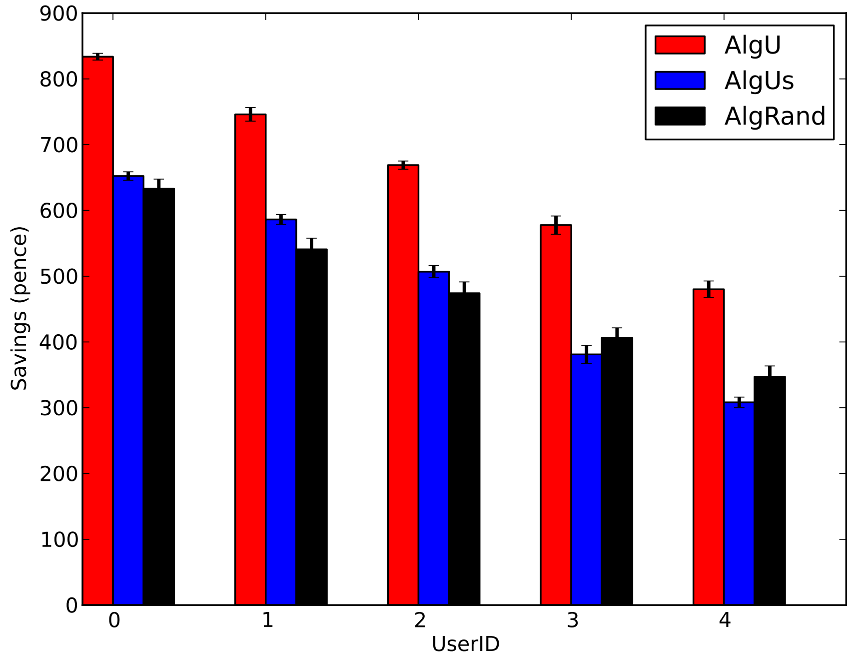

Figure 7.6: Potential savings by selecting the best personalised suggestions within House 1 (REDD dataset).

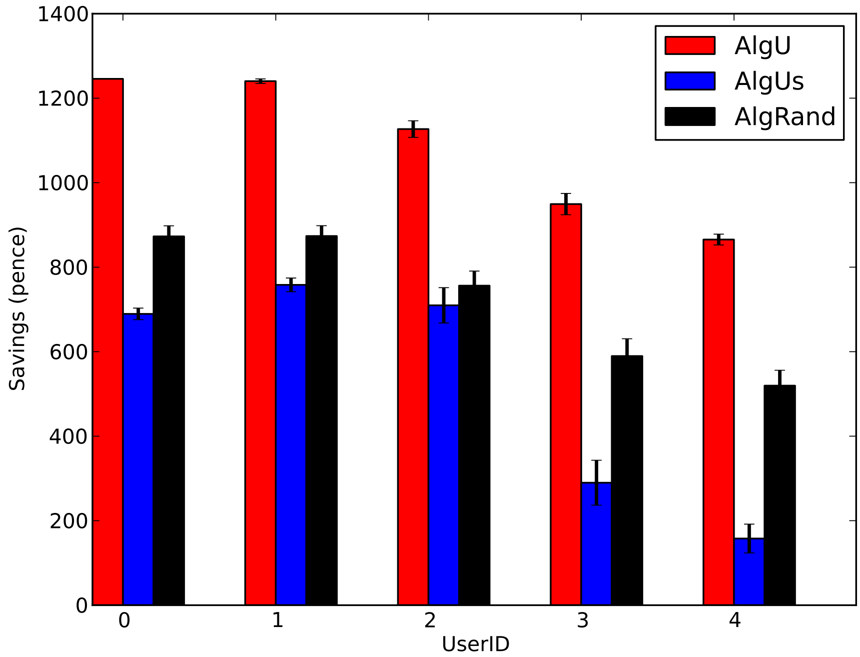

Figure 7.7: Potential savings by selecting the best personalised suggestions within House 3 (REDD dataset).

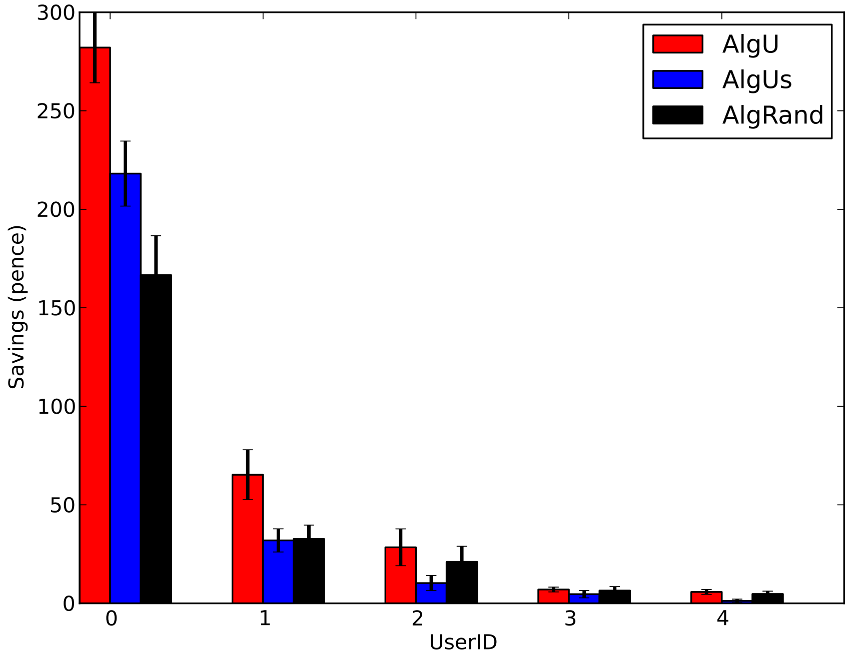

Figure 7.8: Potential savings by selecting the best personalised suggestions within House 4 (REDD dataset).

7.6 Summary

In this chapter, we have proposed the IDR mechanism to improve demand-side management. The IDR mechanism is an extension of the appliance usage prediction problem, allowing the agent to use appliance usage prediction algorithms to interact with a user so that it can learn their preferences to provide personalised feedback. The IDR mechanism has 3 phases: i) Appliance usage prediction (Phase 1), ii) Suggestion optimisation (Phase 2), iii) Human-Agent interaction (Phase 3). Phase 1 makes use of the appliance usage prediction algorithm, GM-PMA, (proposed in Chapter 4). Phase 2 (see Section 7.3) solves a feedback optimisation problem by optimising suggestions to satisfy the user's personal preferences for energy' savings and comfort. Phase 3 (see Section 7.4) describes an algorithm to learn the user's preferences from her responses to the suggestions provided by Phase 2. In addition, a human response model was described in Section 7.2 to provide simulated user behaviours and how decision-making can be processed by the users.

Moreover, we have also evaluated the IDR mechanism by applying it to a variety of simulated human response behaviours within the REDD dataset. We found that, by using IDR, users can always improve their savings. In particular, within a number of our test user profiles, users can save up to £26.23 per week (equivalent to approximately £136.96). We have also evaluated our method (Alg_U) to provide the best *personalised* suggestions that maximises the user's preferences (Phase 2) against two other benchmarks and showed that by optimising the users' personal preferences for comfort and savings can always help to save more money, and up to approximately 90% compared to other benchmarks.

The following chapter provides a detailed summary of the thesis, and and discusses future theoretical and practical implications.

Chapter 8

Conclusions

In Chapter 1, we introduced the key challenges faced in managing energy demand at domestic level. Minimising peaks on the grid can be addressed by DSM techniques, and that a key requirement is to predict energy consumption activities (or appliance usage). We proposed our agent-based DSM approach fundamentally based on solving the appliance usage prediction problem in the smart home. We also identified key requirements that must be fulfilled in order to realise the efficient solution to the problem.

Chapter 2 described the vision of the smart grid, discussed smart meters and the use of DSM techniques. We provided the challenge for standard DSM to manage electricity demand on a large scale and discussed how agent-based approaches could help meet this challenge. We described why agent-based DSM technique that takes the appliance usage prediction problem as a key challenge to be an efficient approach to improve domestic energy management. Then, we highlighted existing works on modelling and predicting inter-dependencies between appliances (or events) in temporal data. In particular, we discussed the two popular approaches that may be used to model the inter-dependencies between appliances: a temporal point process technique, and a rule-based method based on Frequent Episode Discovery. In particular, we used the state–of–the–art algorithms: PCIM (as for temporal point process technique), and EGH (for FED technique) as benchmarks for our evaluation. We showed that as these approaches do not take into account the human routine behaviours and, thus, do not perform well to predict appliance usage on real–world datasets of energy consumption from homes. We then discussed related works for human routine modelling and prediction. We used a popular non-parametric method, Dirichlet Process Mixture, as our benchmark for predicting human routine behaviours. However, this method only deal with one single appliance sequences (i.e, single stream), and does not model the complex inter-dependencies between appliances.

In Chapter 3, we described our appliance usage prediction model formally. The model takes into account historical data (such as the appliance usage) in order to predict when and which appliances will likely to be used in the future. We then described two real–world datasets: the REDD dataset and the FE dataset, that we use to evaluate the prediction performance in this thesis. The REDD dataset was produced using highly intrusive method that monitoring appliance–level energy usage at high granularity and with high accuracy. In contrast, the FE dataset was collected by non-intrusive methods (i.e., house-level monitoring), and therefore involved high uncertainty as appliance usage events were manually tagged by the users who may have mistaken appliance runtime. We also explained in more detail how we extract the raw data from the REDD dataset to use in our prediction model.

In Chapter 4, we described our algorithm, EGH-H, that is designed to take into account the appliance usage dependencies and the human routine and generate the prediction results in a short time (i.e., seconds). In more detail, by considering the weekly cyclic human's routine, EGH-H captures the dependencies between the appliance usage based on identifying the significant patterns (i.e., the rule-based for appliance usage).

Next, in Chapter 5, we described an alternative algorithm, GM-PMA, that captures the human routine and the inter-dependencies between appliance usage. GM-PMA is based on Dirichlet Process Mixture, conditioning on the cyclic human weekly routine to identify sets of latent classes that represent sets of behaviours of the user (e.g., working days, holidays, studying, weekends). These latent classes are hidden, and never directly observed, and hence automatically capture the inter-dependencies between appliance usage at different times of the day.

Then, in Chapter 6. we evaluated the prediction accuracy and runtime of our algorithms: GM-PMA, EGH-H, against other benchmarks (DP, DP-Ext, EGH, and PCIM). We use ROC curves to compare the prediction accuracy of the algorithms. We used the synthetic datasets to evaluate the prediction performance with different settings of inter-dependencies between appliances. We found that GM-PMA dominated other benchmarked algorithms. Then, we empirically evaluated our algorithms and other benchmarks using the two real–world datasets: the REDD dataset and the FE dataset. We found that GM-PMA outperformed other benchmarks up to 41% in term of prediction accuracy. In turn, the runtime of EGH-H is 100 times lower on average, than that of other benchmarked algorithms.

In Chapter 7, we proposed the IDR mechanism in order to extend the use of appliance usage prediction algorithms in the context of demand response to learn user' preferences. In more detail, we evaluated the IDR mechanism to provide insights into how appliance usage prediction algorithms can help to provide personalised suggestions, and hence, learn the user's preferences to minimise discomfort as well as energy costs. We evaluated the IDR on a number of different simulated user behaviours on the REDD dataset based on the measurement of monetary savings. We showed that, by using the IDR, users are likely to improve their monetary savings and that the mechanism can optimise personalised suggestions to users by learning from their responses. We also showed that if we learn the user's preferences (for comfort and savings), up to 90% savings can be achieved compared to other benchmarks that: i) only focus on maximising monetary savings (but ignoring user's comfort), ii) randomly notify the user. Next, we discuss the theoretical (Section 8.1) and practical implications (Section 8.2) of our work.

8.1 Theoretical Implications

Low uncertainty in the use of appliances is a key assumption that we have made in our models. We have showed that our model, GM-PMA, dominates others with regards to the prediction accuracy on the FigureEnergy dataset which has highly uncertain labels (i.e., appliance usage). However, minimising (or correcting) the uncertainty of the appliance usage events can significantly improve the usage prediction accuracy. For example, in our current model of energy consumption activities in the home, we have assumed the appliance usage is correctly annotated (i.e., the information of appliance usage is properly given, such as appliance's types and duration; no data loss occurs). This assumption allowed us to simplify the model of appliance usage prediction, and thus, the prediction algorithm does not have to deal with the uncertainty in the dataset. However, in a real application (e.g., FE), missing appliance usage typically occurs because it is difficult to get users to precisely tag everything they have done in the past. Furthermore, users may mistakenly give the wrong type for the appliance, or energy consumption activity (e.g., cooking instead of having shower), or set the incorrect length for the use of appliance. Hence, the data might be wrong/biased in some cases. Taking into account the uncertainty of the use of appliances will cause difficulties in learning the inter–dependencies of appliance usage, or identifying the pattern of occurrences of the use of appliances. One possible rewards would be to detect the missing usage and fulfill those missing data with some certain level of correct labels as well as correcting the incorrect labelling of the appliance usage. Hence, this fixed dataset may be passed to our appliance usage prediction algorithms.

Another issue is that GM-PMA is only conditioned on the cyclic nature of human routine. To improve the prediction accuracy, there might be others factors that will be highly correlated (e.g., external factors such as weather forecast, different human routines, user's relationship on social network, and the influence of the group's behaviour). GM-PMA can also analyse these factors in more detail. Also, the runtime can be much faster if a online-learning version of GM-PMA can be developed.

8.2 Practical Implications

In this section, we discuss a number of practical implications for the works in this thesis. We enumerate a number of implications that we can pursue for real–world deployment as follows:

- We have evaluated that, by using the IDR mechanism over a variety of simulated human response behaviours, users are likely to improve their savings significantly. A real–world deployment of the IDR mechanism is needed to validate these results.
- We need to design interfaces to incentivise users to shift energy consumption activities and to consider prices when these tend to be low. By so doing, users can be significantly encouraged to trade their convenience of using the appliances in return for savings.
- The "Peer-pressure" approach may be employed as another incentive in addition to potential savings. This approach combines intelligent monitoring of energy use with personalised suggestions and information sharing within user groups (e.g., neighbourhood, social-network groups). Users will be provided, *via* smart meters, their energy consumption patterns compared with those in similar households in terms of size or age.

References

A. Alan, E. Costanza, J. Fischer, S. D. Ramchurn, T. Rodden, and N. R. Jennings. A field study of human-agent interaction for electricity tariff switching. In *Proceedings of the 13th International Conference on Autonomous Agents and Multi-Agent Systems*, page (in press), Paris, France, 2014.

R. Barbieri, M. C. Quirk, L. M. Frank, M. A. Wilson, and E. N. Brown. Construction and analysis of non-Poisson stimulus-response models of neural spiking activity. *Journal of Neuroscience Methods*, 105:25–37, 2001.

C. Berberidis and I. P. Vlahavas. Detection and Prediction of Rare Events in Transaction Databases. *International Journal on Artificial Intelligence Tools*, 16(5):829–848, 2007.

C. M. Bishop. *Pattern Recognition and Machine Learning*, volume 4 of *Information science and statistics*. Springer, 2006. ISBN 9780387310732.

D. M. Blei and M. I. Jordan. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–144, March 2006. ISSN 1936-0975.

E. N. Brown, R. E. Kass, and P. P. Mitra. Multiple neural spike train data analysis: state-of-the-art and future challenges. *Nature neuroscience*, 7(5):456–461, May 2004. ISSN 1097-6256.

G. Casellaa and E. I. Georgeb. Explaining the Gibbs Sampler. *The American Statistician*, 46(3):167–174, 1992.

C. Cho, Y. Wu, S. Yen, Y. Zheng, and A. L. P. Chen. On-line rule matching for event prediction. *The VLDB Journal*, 20(3):303–334, July 2010. ISSN 1066-8888.

C. Chu, T. Jong, and Y. Huang. A direct load control of air-conditioning loads with thermal comfort control. *IEEE Power Engineering Society General Meeting*, 1:664–669, 2005.

E. Costanza, J. E. Fischer, J. A. Colley, T. Rodden, S. D. Ramchurn, and N. R. Jennings. Doing the Laundry with Agents : a Field Trial of a Future Smart Energy System in the Home. In *ACM CHI Conference on Human Factors in Computing Systems 2014*, page (In Press), Toronto, Canada, 2014.

E. Costanza, S. D. Ramchurn, and N. R. Jennings. Understanding domestic energy consumption through interactive visualisation: a field study. In *UbiComp '12 Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 216–225, Pittsburgh, US, 2012. ISBN 9781450312240.

D. J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods*, volume I. 2003. ISBN 0387955410.

S. Darby. The effectiveness of feedback on energy consumption. A review for DEFRA of the literature on metering, billing and direct displays. Technical Report April, 2006.

M Deindl, C Block, R Vahidov, and D Neumann. Load Shifting Agents for Automated Demand Side Management in Micro Energy Grids. *2008 Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, pages 487–488, October 2008.

Department of Energy & Climate Change. Smarter Grids : The Opportunity. (December), 2009a.

Department of Energy & Climate Change. UK National Strategy for Climate and Energy : Transition to a Low Carbon Society. Technical report, 2009b.

Department of Energy & Climate Change. Energy Consumption in the United Kingdom. Technical report, 2010.

N. Eagle and A. S. Pentland. Eigenbehaviors: identifying structure in routine. *Behavioral Ecology and Sociobiology*, 63(7):1057–1066, April 2009. ISSN 0340-5443.

K. Farrahi and D. Gatica-Perez. Discovering routines from large-scale human locations using probabilistic topic models. *ACM Transactions on Intelligent Systems and Technology*, 2(1):1–27, January 2011. ISSN 21576904.

A. Faruqui and S. George. Quantifying Customer Response to Dynamic Pricing. *The Electricity Journal*, 18(4):53–63, 2005.

J. E. Fischer, S. D. Ramchurn, M. A. Osborne, O. Parson, T. D. Huynh, N. Pantidi, S. Moran, K. Bachour, S. Reece, E. Costanza, T. Rodden, and N. R. Jennings. Recommending Energy Tariffs and Load Shifting Based on Smart Household Usage Profiling. In *International Conference on Intelligent User Interfaces*, pages 383–394, 2013. ISBN 9781450319652.

J. Froehlich, L. Findlater, and J. Landay. The design of eco-feedback technology. *Proceeding CHI '10 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1999–2008, 2010.

H. Gao, J. Tang, and H. Liu. Exploring Social-Historical Ties on Location-Based Social Networks. In *Sixth International AAAI Conference on Weblogs and Social Media*, pages 114—-121, 2012.

C. W. Gellings. The concept of demand-side management for electric utilities. *Proceedings of the IEEE*, 73(10):1468–1470, 1985. ISSN 0018-9219.

A. Gomes, C. H. Antunes, and A. G. Martins. A Multiple Objective Approach to Direct Load Control Using an Interactive Evolutionary Algorithm. *IEEE Transactions on Power Systems*, 22(3):1004–1011, August 2007. ISSN 0885-8950.

M C González, C A Hidalgo, and A Barabási. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, June 2008. ISSN 1476-4687.

A. Gunawardana, C. Meek, and P. Xu. A Model for Temporal Dependencies in Event Streams. In *Neural Information Processing Systems Foundation*, pages 1962–1970, 2011.

International Energy Agency. Electricity/Heat in World in 2008. Technical report, 2008.

International Energy Agency. WORLD ENERGY OUTLOOK 2013 FACTSHEET - How will global energy markets evolve to 2035? Technical report, 2013.

H. Ishwaran and L. F. James. Gibbs Sampling Methods for Stick-Breaking Priors. *Journal of the American Statistical Association*, 96(453):161–173, March 2001. ISSN 0162-1459.

C. Jensen, A. Kong, and U. Kjaerulff. Blocking gibbs sampling in very large probabilistic expert systems. *International Journal of Human Computer Studies. Special Issue on Real-World Applications of Uncertain Reasoning.*, 42:647–666, 1995.

A. Kashif, X. H. B. Le, J. Dugdale, and S. Ploix. Agent based framework to simulate inhabitants' behaviour in domestic settings for energy management. In *Proceedings of the 3rd International Conference on Agents and Artificial Intelligence*, pages 190–199, 2011.

J. Z. Kolter and M. J. Johnson. REDD : A Public Data Set for Energy Disaggregation Research. In *In proceedings of the SustKDD Workshop on Data Mining Applications in Sustainability*, number 1, pages 1–6, 2011. ISBN 9781450308403.

S Laxman, P S Sastry, and K P Unnikrishnan. Discovering Frequent Episodes and Learning Hidden Markov Models : A Formal Connection. *IEEE Transactions on Knowledge and Data Engineering*, 17(11):1505–1517, 2005.

S. Laxman, V. Tankasali, and R. W. White. Stream Prediction Using A Generative Model Based On Frequent Episodes In Event Sequences. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2008)*, pages 453–461, New York, USA, 2008. ISBN 9781605581934.

S. N. Maceachern. Estimating normal means with a conjugate style dirichlet process prior. *Communications In Statistics - Simulation and Computation*, 23(3):727–741, 1994.

D. MacKay. *Sustainable energy - without the hot air*. Cambridge University Press, 2007.

K. Mannila, H. Toivonen, and A. I. Verkamo. Discovery of Frequent Episodes in Event Sequences. *Data Mining and Knowledge Discovery*, 1(3):259 – 289, 1997.

J. McInerney, J. Zheng, A. Rogers, and N. R. Jennings. Modelling heterogeneous location habits in human populations for location prediction under data sparsity. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing - UbiComp '13*, pages 469–478, 2013.

OpenHAN Task Force of the Utility AMI Working Group. UCAIug Home Area Network System Requirements Specification. Technical report, 2010.

A. Papoulis. *Probability, Random Variables, and Stochastic Processes.* New York, 3rd edition, 1991.

S. Rajaram, T. Graepel, and R. Herbrich. Poisson-Networks : A Model for Structured Point Processes. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, 2005.

S D Ramchurn, P Vytelingum, A Rogers, and N Jennings. Agent-Based Control for Decentralised Demand Side Management in the Smart Grid. In *The Tenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, pages 5–12, 2011a.

S D Ramchurn, P Vytelingum, A Rogers, and N R Jennings. Agent-Based Homeostatic Control for Green Energy in the Smart Grid. *ACM Transactions on Intelligent Systems and Technology*, 2(4), 2011b.

T. A. Rodden, J. E. Fischer, N. Pantidi, K. Bachour, and S. Moran. At Home with Agents : Exploring Attitudes Towards Future Smart Energy Infrastructures. In *CHI '13 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1173–1182, 2013. ISBN 9781450318990.

N. Ruiz, I. Cobelo, and J. Oyarzabal. A Direct Load Control Model for Virtual Power Plant Management. *IEEE Transactions on Power Systems*, 24(2):959–966, May 2009. ISSN 0885-8950.

F. C. Schweppe, B. Daryanian, and R. D. Tabors. Algorithms for a Spot Price Responding Residential Load Controller. *IEEE Transactions on Power Systems*, 4(2):507 – 516, 1989. ISSN 0272-1724.

J. Sethuraman. A constructive definition of Dirichlet Priors. *Journal of the American Statistical Association*, 4:639–650, 1994.

M. Shimosaka, T. Ishino, H. Noguchi, T. Sato, and T. Mori. Detecting Human Activity Profiles with Dirichlet Enhanced Inhomogeneous Poisson Processes. *The 20th International Conference on Pattern Recognition*, pages 4384–4387, August 2010.

Smart Metering Design Group. Industry s Draft Technical Specifications. In *Smart Metering Implementation Programme*, 2011.

K. Spees and L. Lave. Impacts of Responsive Load in PJM : Load Shifting and Real Time Pricing. *Energy Journal*, 2(29):101–123, 2008.

R. Srikant and Agrawal R. Mining Sequential Patterns. In *Proc. 11th Intl Conf. Data Eng.*, 1995.

G. Strbac. Demand side management: Benefits and challenges. *Energy Policy*, 36(12): 4419–4426, December 2008. ISSN 03014215.

G. Tatavarty, R. Bhatnagar, and B. Young. Discovery of Temporal Dependencies between Frequent Patterns in Multivariate Time Series. *IEEE Symposium on Computational Intelligence and Data Mining*, pages 688–696, 2007.

The Climate Group. Smart 2020: Enabling the Low Carbon Economy in the Information Age, Report by The Climate Group of behalf of the Global eSustainability Initiative (GeSI). Technical report, 2008.

S Tominaga, M Shimosaka, R Fukui, and T Sato. A Unied Framework for Modeling and Predicting Going-Out Behaviour. *Proceedings of the 10th international conference on Pervasive Computing*, pages 73–90, 2012.

N. C. Truong, J. Mcinerney, L. Tran-thanh, E. Costanza, and S. D. Ramchurn. Forecasting Multi-Appliance Usage for Smart Home Energy Management. In *Proceeding IJCAI'13 Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2908–2914, 2013a.

N. C. Truong, J. Mcinerney, L. Tran-Thanh, E. Costanza, and S. D. Ramchurn. Algorithms for Appliance Usage Prediction - Thesis. Technical Report under review, 2014.

N. C. Truong, L. Tran-thanh, E. Costanza, and S. D. Ramchurn. Predicting Energy Consumption Activities for Home Energy Management. In *Agent Technologies for Energy Systems (ATES 2012)*, 2012.

N. C. Truong, L. Tran-Thanh, E. Costanza, and S. D. Ramchurn. Activity Prediction for Agent-based Home Energy Management. In *Agent Technologies for Energy Systems (ATES 2013)*, 2013b.

N. C. Truong, L. Tran-Thanh, E. Costanza, and S. D. Ramchurn. Towards Appliance Usage Prediction for Home Energy Management. *Proceedings of the fourth international conference on Future energy systems*, pages 287–288, 2013c.

US Department of Energy. "GRID 2030" A National Vision for Electricity's Seconds 100 Years. Technical Report July, 2003.

P. Vytelingum, T. Voice, S. D. Ramchurn, A. Rogers, and N. R. Jennings. Theoretical and Practical Foundations of Large-Scale Agent-Based Micro-Storage in the Smart Grid. *Journal of Artificial Intelligence Research*, 42:765–813, 2011.

P. Vytelingum, T. D. Voice, S. D. Ramchurn, A. Rogers, and N. R. Jennings. Agent-based Micro-Storage Management for the Smart Grid. In *The Ninth International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, pages 39 – 46, Toronto, Canada, 2010.

D. D. Weers and M. A. Shamsedin. Testing a New Direct Load Control Power Line Communication System. *IEEE Transactions on Power Delivery*, (3):657–660, 1987.

G. M. Weiss and H. Hirsh. Learning to Predict Rare Events in Event Sequences. In *In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pages 359–363, 1998.

V. W. S. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia. Autonomous Demand-Side Management Based on Game-Theoretic Energy Consumption Scheduling for the Future Smart Grid. *IEEE Transactions on Smart Grid*, 1(3):320–331, 2010.

R. Yang and M. W. Newman. Learning from a Learning Thermostat : Lessons for Intelligent Systems for the Home. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing (UbiComp 2013)*, pages 93–102, 2013. ISBN 9781450317702.

F. Ygge and H. Akkermans. Decentralized Markets versus Central Control: A Comparative Study. *Journal of Artificial Intelligence Research*, 11:301–333, 1999.

F. Ygge, H. Akkermans, A. Andersson, M. Krejic, E. Boertjes, E. Ab, and C. Teknikpark. The HOMEBOTS System and Field Test: A Multi-Commodity Market for Predictive Power Load Management. In *In Proceedings of the Fourth International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agents (PAAM99)*, pages 363–382, 1999.