Leveraging Submetered Electricity Loads to Disaggregate Household Water-use

by

Bradley Ellert

B.Sc., University of Lethbridge, 2013

Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science

in the
School of Computing Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Summer 2015

Approval

Name:	Bradley Ellert
Degree:	Master of Science (Computing Science)
Title:	*Leveraging Submetered Electricity Loads to Disaggregate Household Water-use*
Examining Committee:	**Dr. Nick Sumner** (chair) Assistant Professor

Dr. Fred Popowich
Senior Supervisor
Professor

Dr. Wolfgang Stuerzlinger
Supervisor
Professor
Interactive Arts and Technology

Dr. Jian Pei
Internal Examiner
Professor

Date Defended: August 17, 2015

Abstract

The world's fresh water supply is rapidly dwindling. Informing homeowners of their water-use patterns can help them reduce consumption. Today's 'smart' meters only show a whole house's water consumption over time. People need to be able to see where they are using water most to be able to change their habits. The task of inferring the breakdown of water-use from smart meter data is called water disaggregation. Water disaggregation has been dominated by studies that rely on high-frequency data, proprietary meters, and/or labelled datasets. In contrast, this thesis uses low-frequency data from standardized meters and does not rely on labelled data. To accomplish this, we leverage information from non-intrusive load monitoring, the electricity counterpart of this task. We propose a modification of the Viterbi Algorithm that applies a supervised method to an unsupervised disaggregation problem. Using this, we are able to achieve mean squared errors of under 0.02 L^2/min^2.

Keywords: water disaggregation; water conservation; smart homes; sustainability

Dedication

To my parents, for always being there to listen.

Acknowledgements

I would like to thank the many professors whose influence led to this thesis. Dr. Nicole Rosen sparked my interest in research at the University of Lethbridge. Dr. Inge Genee continued to fuel my passion for Linguistics throughout my undergraduate degree. Dr. Jacqueline Rice provided me a 'mini thesis' experience by means of an independent study. Dr. Kevin Grant gave me personal advice about continuing on with graduate studies. Dr. Anoop Sarkar saw me through many shifts in my research topic and eventually supported my change of supervisory committee to a field separate from Natural Language Processing. Dr. Fred Popowich took on the role as my senior supervisor and helped me flesh out a research topic in the field of Computational Sustainability, an area entirely new to me. Dr. Wolfgang Stuerzlinger joined my supervisory committee on good faith, having just met me. Dr. Jian Pei acted as my examiner, remembering me from his Data Mining course. Dr. Nick Sumner agreed to chair my thesis defence on quite short notice.

Most importantly I must stress the thanks that goes out to Dr. Stephen Makonin. Though not part of my committee, Dr. Makonin's advice was invaluable. My thesis would not have been possible without the foundation laid out by his dissertation. Having direct insight into AMPds from the author himself helped me focus on my research questions.

This degree was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC), SFU's Dean of Graduate Studies (DGS), and SFU's Work Integrated Learning (WIL). Thank you to NSERC for awarding me the Alexander Graham Bell Canada Graduate Scholarship. Thank you to DGS for awarding me Provost's Prize of Distinction, the C.D. Nelson Memorial Graduate Scholarship, and a Graduate Fellowship. Thank you to WIL, particularly Greig Baird, for helping me land a co-op position at ODGS.

Last but not least, personal thanks must be given those who helped me maintain my sanity. My family has been tremendously supportive of my pursuit of this degree. Having my mom's family in the Lower Mainland has made my moves to Burnaby and New Westminster so much smoother. It is always great to be able to visit my dad's family in Alberta on no notice and be welcomed in. Both my parents and Jocelyn, my sister, listened to hours of my worries over the phone. I deeply appreciate friends from as far back as grade school and my time working at Safeway keeping in touch. And finally many thanks to Lindsey, my girlfriend, for bearing the brunt of my craziness and sticking with me through it.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The world's fresh water supply is rapidly dwindling. Water consumption has nearly doubled worldwide since 1950 [45]. While demands are increasing due to population growth and loftier standards of living, supplies are decreasing due to pollution and unsustainable agricultural practices [15]. Projections of future water-use foresee this trend continuing. They range from steady increases that will catch up to us over time to drastic spikes that will leave some communities without water in the near future [23].

Looking at Canada, the majority of the population being in the south puts a strain on a concentrated portion of the country's fresh water supply. Since at least the 1970s, this area exhibited a downward trend in water yield. In particular, the Prairies (with a lower than average yield due to the drier climate) and Southwestern Ontario (with a higher than average intake due to the denser population) use more than 40% of the available water [41]. It has been shown that volume-based water rates encourage residents to use less water than those on flat rates. Despite this, almost 30% of households in Canada are not even metered [14]. Communities are just beginning to experiment with 'smart' meters.

These smart meters only address part of the problem of helping homeowners use water wisely. By themselves, they only show a whole house's water consumption over time. In addition, people need to be able to see where they are using water most to be able to change their habits. Environmental psychology studies have shown an average decrease in consumption of 18% after introducing eco-feedback [19]. Groups have already begun experimenting to see how users react to different types of data visualizations [20].

Before this breakdown can be reported in a real-world setting, unless water consumption is monitored for individual devices and appliances, a method must exist to infer this information from smart meter data. This task is called water disaggregation. Given a whole house water meter reading, we want to separate it into its component uses. Many of the existing systems require expert tuning, difficult installation, and/or high-frequency data making them infeasible for broad application (Chapter 2). By the frequency of the data,

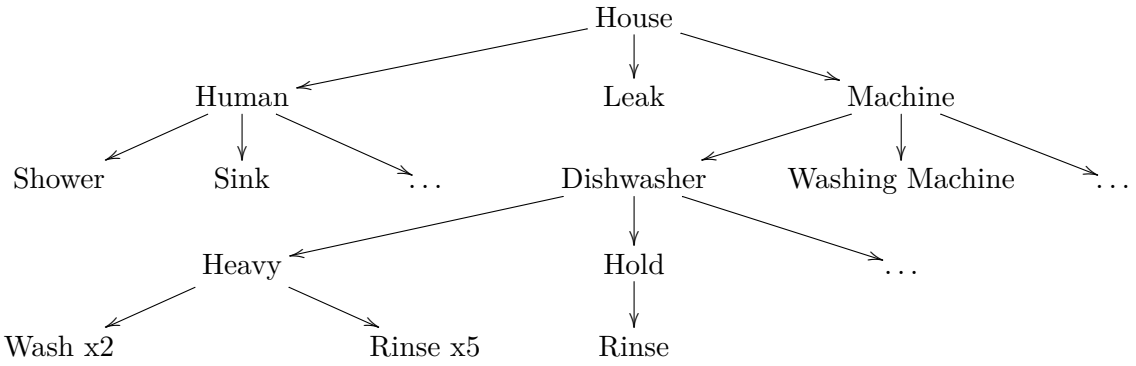

Figure 1.1: Household water-use can be broken down by agent, fixture, activity, and action.

we refer to the rate at which readings are taken. *High-frequency* data refers to a sample rate between 1 Hz and 1 MHz [6].

Recent studies have focused on differences between fixtures that are similar to each other (e.g. kitchen sink versus bathroom sink). We do not think this is very informative in terms of motivating water conservation. It is more important for users to know how different types of fixtures (e.g. showers versus dishwashers) contribute to the total water reading. The focus on individual fixture-level disaggregation is more appropriate in an activity monitoring setting. Water monitoring can also help us infer home occupant locations [16, 6]. This information could be used in elder care applications.

Our focus is on taking advantage of data that is more likely to be available in a real-world water conservation setting. This means using *low-frequency* data that can be gathered from commodity hardware, creating the potential for real-time results. Less data collected also means increased privacy. We restrict ourselves to extracting water-use by machines by leveraging information from non-intrusive load monitoring (NILM), the electricity counterpart of this task [25]. Previous methods tend to rely on heuristics for these and focus rather on human-use. Once machine-use has been determined, the remaining data on human-use can be used to show residents where they can make changes to save water.

1.1 A Hierarchical View

Household water consumption can be viewed as a hierarchy (Figure 1.1). In the broadest sense, it can be broken down based on the agent causing the consumption: *human*, *machine*, and *leaks*. Human-use refers to fixtures that can be used for variable lengths of time with varying flow rates, such as showers and sinks. Machine-use refers to appliances that follow cycles with fixed patterns after being initiated, such as (clothes) washing machines and dishwashers. More specifically, machine water-use occurs in devices where water and

electricity consumption can be correlated. Leaks refer to any water loss ranging from broken pipes to dripping faucets. Our work focuses on finding events of the machine-use variety, as they can be correlated with energy consumption.

Different instances of these events can be further categorized into types based on different machine settings. For example, the dishwasher may be programmed to do a *heavy wash* or to *hold* the dishes. We adopt the term *activity* from the field of ambient intelligence and pervasive computing to describe these event types. Using a paradigm from this field, we recognize activities as being composed of *actions* [42]. In the case of the dishwasher, these actions could be *wash* and *rinse*. A heavy wash might be made up of two washes followed by five rinses, while a hold simply rinses the dishes once to keep them fresh. The same principle could be applied to the washing machine.

Certain household water-users can be more difficult to classify. For example, toilets could be considered machines as they are constrained to consuming fixed amounts of water. However since we are defining machines as devices where water and electricity consumption can be correlated, toilets fall under human-use. One exception to this is the Japanese 'Washlet' which includes an electronically motored bidet [7]. Western toilets would typically be limited to *low flush* and *high flush* activities (if these options are available).

Another hard to classify category is irrigation. Sprinklers are often put on timers, which are sometimes mechanical. Even if watering times cannot be derived from the house's electrical information, this data can be gleaned from other sources. For example, many urban centres restrict lawn watering to certain times of the day [9] (known as 'hosepipe bans' in the United Kingdom [5]). Approaching the problem from the other side, disaggregating this data could help with the enforcement of outdoor water-use restrictions.

Note that given our focus and our activity paradigm, fixture types such as sink are not broken down into specific fixture instances. A typical activity here would simply be composed of a single action. Given low-frequency data, the variable flow rate during the opening and closing of the tap will not register and cause issue. However, this action will be of variable length. For this reason, and since these activities cannot be correlated with electrical consumption, human-use falls outside the scope of this thesis.

1.2 Our Approach

The main contribution of this thesis is an algorithm for disaggregating machine water-use from main water meter data, given disaggregated electricity data. This allows low-frequency, unlabelled data to be used to build a models specific to each household's appliances. Domain knowledge about general consumption patterns is not needed, as this information is learned from smart meter data. By not relying on generic ontological information, the system does not make incorrect assumptions about different makes and models of appliances and is able to be used on future machines that may follow unforeseen patterns.

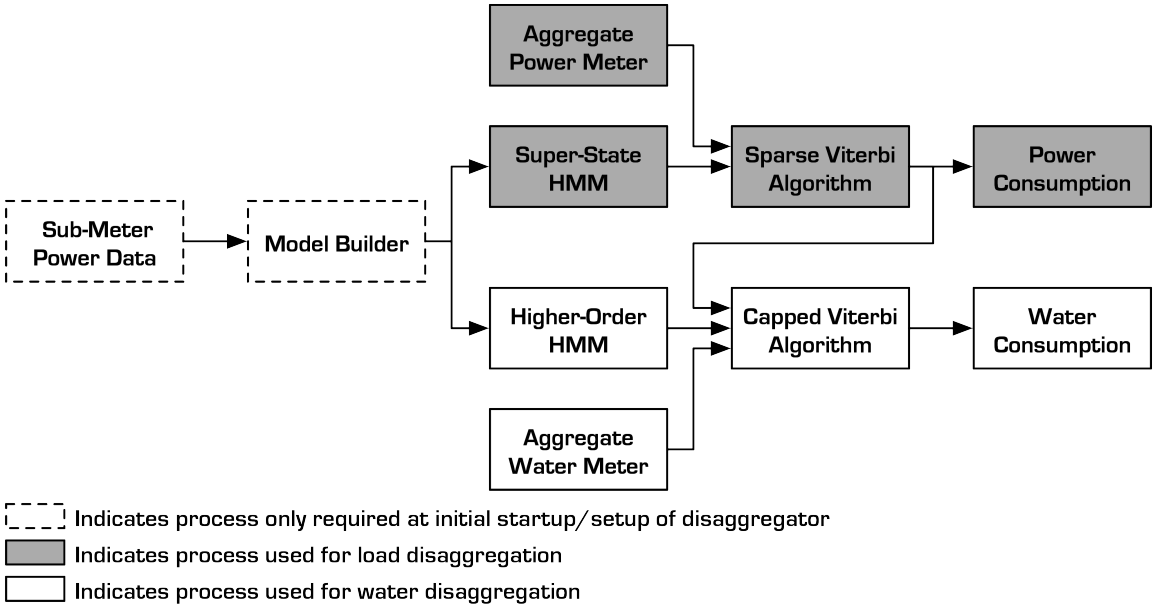

Figure 1.2: A block diagram depicting how water disaggregation can be paired with an NILM (load disaggregation) process. NILM appliance classification is used by the water disaggregator to disaggregate that appliance's water consumption.

Our approach represents the first water disaggregator that has the potential of providing homeowners with detailed appliance water-use information outside of a controlled setting without the need for tuning by a trained professional or the installation of non-standard metering equipment. By developing this technology alongside NILM techniques, we open the door to a symbiotic relationship between these common related tasks, instead of relying on more obscure sensor data. Figure 1.2 shows an outline of such a system.

To evaluate our method, we use the Almanac of Minutely Power Dataset (AMPds) [33]. In doing so, we produced a hand-labelling of the dishwasher's water-use. We provide a quantitative analysis of our system's performance on the dishwasher and a qualitative analysis of its performance of the washing machine. Not finding any other suitable datasets for this kind of evaluation, we show the need for standardized datasets like AMPds.

Presented here is an outline of the format of this thesis. Chapter 2 provides a literature review summarizing the previous work in the field of water disaggregation. It highlights how the focuses of these methods differ from this work. Chapter 3 details our procedure. A concrete problem definition is stated at the start of §3.1. Here AMPds and a method for load dissaggregation (comprising the grey boxes in Figure 1.2) are introduced. We describe the process of extracting activities in §3.2 and our model (comprising the white boxes in Figure 1.2) is laid out in §3.3. In Chapter 4 we show experiments performed to evaluate our model. Finally, Chapter 5 summarizes the results of our research and points out possible future directions.

Chapter 2

Previous Work

The field of water disaggregation is still quite young, and therefore relatively few papers have been published on the topic. Table 2.1 summarizes the previous work in the field of water disaggregation. Initial methods required time-consuming hand-tuning by experts for individual households. This is only practical for small-scale studies to gather samples for assessing regional water-use.

Newer methods targeted at homeowner use still have barriers to adoption. They often require expensive labels for training data and make assumptions about general water consumption patterns. A lack of a standardized dataset makes the results of different systems difficult to compare. Additionally, a wide range of proprietary meters are used that take measurements in different units. Most of these rely on high-frequency sampling.

2.1 Flow Trace Analysis

The first true method for water disaggregation was developed by Aquacraft, a water management company based in Boulder, Colorado [11]. Their Trace Wizard performs flow trace analysis using a largely manual process. A flow trace is simply a plot of water flow rate over a period of time. This information was collected using a retrofit device that attaches to a water meter and logs flow rate every 10 seconds by measuring changes in magnetic field. Newer 'smart' meters provide this information natively [35].

For their original study, 16 participating homes were selected from a neighbourhood outside the city. First, signature traces were collected for each fixture and appliance in each house. With these patterns in mind, technicians were able to label future examples by hand. To simplify this process, a set of heuristics were developed to automatically categorize based on the flow rate and duration exemplified in the larger set of self-consistent examples. The details of this algorithm are not disclosed. Dishwashers and washing machines were noted as difficult to distinguish because they often co-occurred with miscellaneous faucet use.

Name	Group	Dates	Contribution	Measurement	Algorithm
Trace Wizard	Aquacraft	1996 - 2004	Seminal papers	Flow rate	Manual labelling
NAWMS	UCLA	2008	Flow rate estimate	Vibration	Linear programming
WaterNILM / WaterWOLF	MIT	2014	Single sensor	Vibration	LDA / QDA
HydroSense	University of Washington	2009 - 2014	Consumer feedback	Pressure	Template classifier
WaterSense / FixtureFinder	University of Virginia	2011 - 2013	Motion detectors	Flow rate	Bayesian network
SCFDD / DSCRDM	Virginia Tech	2011 - 2013	Low sample rate	Flow rate	DDSC

Table 2.1: The major previous works in water disaggregation.

An oft-cited study by the same group expands the scope to 1,188 homes across 12 sites [34]. The results of flow trace analysis on this dataset have been used as ground truth by others for machine learning approaches. Accuracy is only maintained by periodic hand checks on subsets of the data since no submeters are actually installed for verification.

Trace Wizard has been used in many different studies. Previous literature reviews have focused on summarizing the application of this and related flow trace analysis tools [31]. Flow trace analysis is limited to determining the fixture class (e.g. toilet, sink) of an instance of water-use. Recently, the trend has been to focus on finer grained disaggregation down to the level of specific fixture (e.g. kitchen sink, bathroom sink). The rest of this literature review focuses on these advances.

2.2 Vibration Based Methods

NAWMS (Nonintrusive Autonomous Water Monitoring System) is the first method to demonstrate disaggregation down to the level of individual pipes [29]. The authors show that flow rate can be estimated based on pipe vibration.

Data was collected at a frequency of 100 Hz from their own plumbing test bed consisting of 3 pipes of varying materials and diameters connected to a water main. Values were recorded for the flow rate of the water main and the vibration of the each pipe with individual accelerometers. A cubic root curve was fit to estimate the flow rate $f(t)$ from the vibration $v(t)$ at each time t:

$$f(t) = \alpha \sqrt[3]{v(t)} + \beta \sqrt{v(t)} + \gamma v(t) + \delta.$$

They modelled the problem using linear programming [2] where the task is to minimize the difference between the true flow rate of the water main $M(t)$ and the sum of the estimated

flow rates $\sum_{i=1}^{N} f_i(t)$ for each of the N sub-pipes indexed by i. Defining for K samples

$$\mathbf{M} = [M(1), M(2), \ldots, M(K)]^T$$
$$\mathbf{F} = \left[\sum_{i=1}^{N} f_i(1), \sum_{i=1}^{N} f_i(2), \ldots, \sum_{i=1}^{N} f_i(K)\right]^T,$$

the optimization problem is

$$\begin{aligned} \min \quad & ||\mathbf{M} - \mathbf{F}||_1 \\ \text{subject to} \quad & 0 \leq f_i(t), \end{aligned}$$

where $||\mathbf{M} - \mathbf{F}||_1 = \sum_{t=1}^{K} |M(t) - \sum_{i=1}^{N} f_i(t)|$ is the ℓ_1 norm.

The estimated flow rate of each pipe was compared to the true value monitored by submeters used only for evaluation purposes. Though the average error in estimated flow rate could be as much as almost 5% with up to nearly 10% standard deviation, the accumulated consumption remained accurate with just over 1% error. These results only serve as a proof of concept, as this approach does not physically scale up. This set of sensors is only designed assuming the water main is only split into a simple series of sub-pipes. More realistic scenarios would have sub-pipes split into sub-sub-pipes. Homeowners cannot be expected to install dozens of sensors in their houses. For a typical 3-bedroom 2-bathroom house, the authors estimate 20 sensors would be required. In an apartment building, the number would be much greater than this.

More recently, WaterNILM (Non-Intrusive Load Monitoring) applied a similar physical architecture to a real-world setting [38]. Here, the authors developed a technique requiring only two accelerometers for an entire house. In three test homes, accelerometers were installed downstream from the water meter and at the outlet of the hot water tank.

The accelerometers gather data at a frequency ranging from 12 kHz to 16 kHz. These readings are down-sampled to 4 kHz and the stream is segmented into 0.75 second chunks. Features are extracted from these. Training segments labelled with a load ID are clustered using linear discriminate analysis (LDA) and quadratic discriminate analysis (QDA) [26]. New segments are labelled with a load ID by being classified into a cluster.

In QDA, the class k that maximizes

$$\delta_k(x) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) + \log\pi_k$$

where Σ_k is the covariance matrix, μ_k is the centroid, and π_k is the prior probability is chosen for input features x. For LDA, it is assumed that $\forall k, \Sigma_k = \Sigma$. The discriminate score is simplified to

$$\delta_k(x) = x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + \log\pi_k$$

To test the method, only the predicted load ID is compared to the ground truth. Using QDA and data from both accelerometers was shown to produce the best results, with a misclassification rate under 2%. Heavy emphasis is placed on their WaterWOLF system [37] which uses magnetic fields to estimate flow rate, an already standard procedure when it comes to smart meters. Addressing simultaneous water-use requires training examples of each possible combination in question. Even with this, true disaggregation is not performed, as only the combined label is identified as the source of the total water consumption at such a point.

2.3 Pressure Based Methods

The most active research group in the field of water disaggregation has been the developers of HydroSense [21]. HydroSense introduces a simple single point sensor for whole house water pressure that takes readings at a rate of 1 kHz. This sensor is attached to an unused tap (most commonly outside) and used to approximate flow rate.

For the initial trial, labelled data was collected from ten homes in four cities. In each home, the baseline static water pressure was measured and pressure signatures were taken for each valve (hot and cold) on each fixture using their proprietary HydroSense unit. Toilets were logged both in terms of flushes and full fill cycles. Partially opened faucets were not recorded. In four of the ten homes, flow rate information was collected for faucets and showers by measuring the amount of time needed to fill a one gallon bucket. Five trials were conducted for each pressure signature and flow rate value.

A sliding window of 1000 samples (1 second) is used to analyze the pressure signal. Heuristic techniques are used to extract valve (open/close) events. Valve events are defined to start when either (a) the derivative exceeds 2 psi/sec, or (b) the maximum change in pressure within the sliding window exceeds 1 psi. These thresholds are based on a home with a static pressure of 45 psi and must be scaled accordingly. The end of a valve event is the first point where the fluctuating pressure is less than 5% its magnitude at the beginning of the event. These are classified as open or close events based on either (a) the change in pressure from the start to the end of the event exceeding 2 psi (a decrease for opens and an increase for closes), or (b) or the sign of average derivative at the start of the event (positive for opens and negative for closes).

The valve events are associated with individual fixtures by their similarity to other events ('templates') in the same home based on four distance metrics. Only templates that pass all four metrics with a fixed similarity threshold are considered. Similarity thresholds are learned from the other homes. If no template passes all four filters, the event is left unlabelled. On the other hand, if templates from different fixtures pass, the best is chosen using a nearest-neighbor classifier.

10-fold (home-by-home) cross-validation was used to train a classifier for the (isolated) fixture open and close events and then leave-one-out cross-validation was used to measure classification accuracy within each home. In the homes where actual flow rate was measured, one of the five trials was used to guide an inference of flow rate for the other four trials in another cross-validation setting. The average error rate for each home is reported. This value was usually around 5%, but was shown to be very sensitive to the location where the pressure sensor was installed.

Further enhancements have been developed to produce a self-powered HydroSense unit [3]. More recently, an extension to their system has reduced the amount of training data required by introducing a consumer calibration aspect that queries the user to label difficult examples [30]. An extended analysis using staged experiments was also performed [31].

Despite these advancements, the process still requires the installation of a pressure sensor that renders one valve (typically outdoor) unusable. Labelled training data is needed from other homes and from within the same home, making the classification a very expensive procedure.

2.4 Leveraging Additional Environmental Information

Other than the original flow trace analysis, the methods mentioned up to this point have relied on sensors that monitor attributes of a house's plumbing besides the direct flow rate and total water consumption. Two series of publications ([39, 40] and [44, 12]) initiated more recently have brought the focus back to looking directly at the flow rate readings provided by today's standard smart water meters. The rest of this literature review details these studies.

In particular, this section focuses on works that leverage information from additional environmental sensors to help with predictions. One such method, WaterSense, utilizes data captured by motion sensors to help with water disaggregation [39]. While other machine learning approaches are supervised and require days of training data, this one claims to be unsupervised.

Their system was deployed for a week in two homes. In addition to water main readings at a frequency of 2 Hz, readings were taken once every 7 seconds from passive infrared (PIR) motion detectors in three rooms with water fixtures (two bathrooms and kitchen). Ground truth sensors were installed on the kitchen sink and the toilet and sink in each bathroom for evaluation purposes only. Contact switches report when a fixture is using water.

First, water-use samples are extracted using Canny edge detection [4]. A simple Bayesian network (Figure 2.1) is used to cluster the samples into rooms based on the temporal proximity of motion sensor readings. Here W_i is the sample, c_i is the cluster it belongs to based on duration and flow rate, D_i is the temporal distance to the motion sensors, and r_i

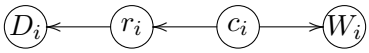

Figure 2.1: The Bayesian network used in WaterSense.

is the room. The room is determined by

$$\hat{r_i} = \arg\max_{r_i} \sum_{c_i} P(r_i|c_i) \cdot P(W_i|c_i) \cdot P(c_i) \cdot P(D_i|r_i)$$

The initial c_i are formed using a technique from genomics [27]. Toilets are differentiated from sinks simply by having an average flow rate greater than 0.3 kL/hour and a duration greater than 30 seconds. Scores between 80% and 90% accuracy are reported. These metrics are only comparing each device's total consumption over the course of a week.

This approach only focuses on the human side of our hierarchy defined in §1.1. The machine side is left to flow trace analysis. Here the authors place too much of an emphasis on disaggregating separate instances of the same type of device, while assumptions about general consumption patterns are still made.

Though WaterSense reduces the need for additional water monitoring devices, it requires the installation of multiple motion sensors. An assumption is made that the proliferation of ubiquitous computing will spawn an abundance of such devices with readily available information. As things stand, we are not at the point where we can rely on this. The authors are already looking toward using non-binary PIR sensors for motion information to achieve more accurate results when multiple rooms are occupied simultaneously.

Building off of this foundation, the same authors created FixtureFinder [40]. FixtureFinder is able to perform the same task as WaterSense without knowing beforehand the number and type of fixtures in each room. At the end, these clusters would still need to be labelled by a human.

The major gain here is that they provide a more generalized framework and demonstrate how it can also be applied to detecting light fixtures. They allude to the possibility of leveraging both water and energy information together, but have not yet attempted this.

2.5 Restricting to Lower Frequency Readings

All of the aforementioned methods rely on high-frequency sensor information (at least once per second). This makes it possible to detect rising and falling edges in the data and isolate periods of consistent readings. Corresponding to a real-world setting with commodity hardware and privacy concerns, it makes sense to restrict oneself to lower frequency data streams [28]. The problem becomes much more difficult with less consistency due to the lower granularity.

SCFDD (Sparse Coding with Featured Discriminative Dictionary) presents a novel way to represent a device's water consumption trend [44]. For each example of water-use by the fixture, the series of time segments is transformed to a binary sequence based on whether the water consumption is more or less than the previous segment. These are used to construct basis functions in a dictionary for each device.

By using an extremely low sampling rate of once every 15 minutes, each example can be an entire day. In this way, the time of day is automatically encoded. This prevents the opportunity for developing a real-time system. Here the F-measure (the harmonic mean of precision, the ratio of correct to predicted, and recall, the ratio of correct to ground truth) can be used to compare exact values, since the granularity is so coarse.

To test the method, flow trace data from an Aquacraft study was used as labelled ground truth [34]. Only three devices were tested: toilets, showers, and washing machines. F-measures of around 65% were reported for toilets and showers. Washing machines were more difficult with an F-measure of around 30% reported.

In an attempt to increase accuracy, the authors also developed DSCRDM (Deep Sparse Coding Based Recursive Disaggregation Model) [12]. They noted that the problem could be reformulated as a non-convex mixed integer nonlinear programming problem instead of relying on an alternative representation of the data. However, this problem is NP-complete. To overcome this, they approximate the result by iteratively decomposing the aggregate water reading one device at a time. After the first device has been separated, the second device is disaggregated from the residual, and so on until one device remains. This reduces the time complexity from $O(k!)$ to $O(k^2)$.

Using the same Aquacraft dataset, they were able to improve the shower F-measure to over 70% and the washing machine F-measure to over 45%. The toilet F-measure decreased to around 35% which was not commented on. They were also able to test many more fixture types without the need to come up with hand-crafted dictionaries of sparse codings for each device's possible consumption patterns as in the previous method.

Though these methods more accurately depict real-world scenarios than WaterSense and FixtureFinder with low-frequency data, the tradeoff is that they require labelled data. Sampling at such a low frequency means that the results are more useful for seeing overall consumption instead of specific patterns while a device is being used.

2.6 Summary

Water disaggregation has been dominated by studies that rely on high-frequency data, proprietary meters, and/or labelled datasets. The focus has been on teasing out subtleties in human-use before more fundamental parts of the problem have been solved. In contrast, this thesis uses low-frequency data from standardized meters and does not rely on labelled data to identify machine water-use from household water data.

Chapter 3

Our Approach

This chapter summarizes our procedure for disaggregating household machine water-use. A concrete problem definition is stated at the start of §3.1. Here we introduce the main dataset used for this study [33]. We describe the process of extracting activities from the dataset in §3.2. Our learning model is detailed in §3.3.

3.1 Our Problem

We leverage data from electricity disaggregation to help with water disaggregation. Given the electrical state of a water-consuming appliance and the whole house water meter reading, the goal is to build a model that can predict the amount of water used by the appliance.

3.1.1 Disaggregating Electricity

In [32], a method for NILM is provided for the electricity domain. By quantizing the electricity readings of an appliance based on peaks in its probability mass function, the time series can be viewed as a list of discrete state transitions. Current (amperes) is shown to be a better measure for determining these states than voltage (volts) or power (watts). These states may correlate to different actions of the appliance. For example, in the dishwasher, one may represent the electricity consumed when the water pump is on, while another may represent the electricity consumed when the heating element is on. These electricity-use actions are distinct from, but related to, the water-use actions of the appliance. Note that these states may be compositional.

For our purposes, we assume we have a method for obtaining the series of states an appliance transitions through, given the series of whole house electricity readings. The method described in [32] determines the entire house's superstate (a combined state representing the state of each appliance simultaneously). A Hidden Markov Model (see §3.3) is built that takes the whole house's electricity reading as input and produces the house's superstate as output. To test our model, we just use the submetered ground truth data.

3.1.2 Our Dataset

To build and test our model we used AMPds (The Almanac of Minutely Power Dataset) [33]. AMPds is a standardized, low-frequency dataset capturing meter information from a household in the Greater Vancouver region of British Columbia, Canada. It is designed with disaggregation in mind, providing data for electricity, water, and natural gas. Compared to other datasets for this purpose, this data is of extremely high quality and has been cleaned to ensure that different researchers will be working from the same starting point.

The dataset contains two years of data from the beginning of April 2012 to the end of March 2014. Meters read in at a rate of once per minute. This means there are over one million records for each meter; a large dataset for low-frequency data. In addition to the main electricity meter there are 20 submeters, resulting in over 20 million total electricity records. For water there are only meters for the main and the instant hot water unit, resulting in over two million total water records. There are also meters for the natural gas main and the furnace, resulting in over two million total natural gas records.

The electricity readings are taken by two DENT PowerScout 18 units [13]. Each record includes measurements for voltage (nearest tenth volt), current (nearest tenth ampere), frequency (nearest hundredth hertz), displacement power factor (nearest hundredth), apparent power factor (nearest hundredth), real power (nearest watt), real energy (nearest watt-hour), reactive power (nearest volt-ampere reactive), reactive energy (nearest volt-ampere-hour reactive), apparent power (nearest volt-ampere), and apparent energy (nearest volt-ampere-hour). This means there are over ten million data points for each electricity meter, resulting in nearly 250 million electricity data points total.

3.1.3 Water Information

Elster/Kent V100 water meters are used to take the water readings [10]. Each record includes a pulse counter (litres), the average rate (litres per minute), and the instantaneous rate (litres per minute). This means there are over three million data points for each water meter, resulting in over six million water data points total. For our purposes we only need the average flow rate, which can also be determined directly from the change in the pulse counter, since we are measuring in L/min and pulses are recorded once per minute.

Water readings are collected in half-litre pulses. For the first few months of AMPds, the water meters were only set to pulse at every gallon (3.785 L). Due to this, we mainly consider the second year of AMPds in this study. Table 3.1 shows an excerpt of the water readings at the beginning and end of AMPds.

Since the water data is not submetered, there is no ground truth to test against. To mitigate this, the author hand-labelled the second year of the dataset. This data is not needed to train the model and is only used for testing purposes.

TIMESTAMP (yy/mm/dd hh:mm)	COUNTER	AVG RATE	INST RATE
1333263600 (12/04/01 00:00)	0.000	0.000	0.000
1333263660 (12/04/01 00:01)	3.785	3.785	0.000
1333263720 (12/04/01 00:02)	3.785	0.000	0.000
1333263780 (12/04/01 00:03)	3.785	0.000	0.000
1333263840 (12/04/01 00:04)	7.570	3.785	0.000
1333263900 (12/04/01 00:05)	7.570	0.000	0.000
1333263960 (12/04/01 00:06)	7.570	0.000	0.000
1333264020 (12/04/01 00:07)	7.570	0.000	0.000
1333264080 (12/04/01 00:08)	7.570	0.000	0.000
1333264140 (12/04/01 00:09)	7.570	0.000	0.000
1333264200 (12/04/01 00:10)	7.570	0.000	0.000
1333264260 (12/04/01 00:11)	7.570	0.000	0.000
1333264320 (12/04/01 00:12)	7.570	0.000	0.000
1333264380 (12/04/01 00:13)	7.570	0.000	0.000
1333264440 (12/04/01 00:14)	7.570	0.000	0.000
1333264500 (12/04/01 00:15)	7.570	0.000	0.000
1333264560 (12/04/01 00:16)	7.570	0.000	0.000
1333264620 (12/04/01 00:17)	7.570	0.000	0.000
1333264680 (12/04/01 00:18)	7.570	0.000	0.000
1333264740 (12/04/01 00:19)	7.570	0.000	0.000
1396334400 (14/03/31 23:40)	548874.474	0.000	0.000
1396334460 (14/03/31 23:41)	548874.474	0.000	0.000
1396334520 (14/03/31 23:42)	548874.474	0.000	0.000
1396334580 (14/03/31 23:43)	548875.474	1.000	0.000
1396334640 (14/03/31 23:44)	548881.974	6.500	8.571
1396334700 (14/03/31 23:45)	548881.974	0.000	0.000
1396334760 (14/03/31 23:46)	548881.974	0.000	0.000
1396334820 (14/03/31 23:47)	548881.974	0.000	0.000
1396334880 (14/03/31 23:48)	548881.974	0.000	0.000
1396334940 (14/03/31 23:49)	548881.974	0.000	0.000
1396335000 (14/03/31 23:50)	548881.974	0.000	0.000
1396335060 (14/03/31 23:51)	548881.974	0.000	0.000
1396335120 (14/03/31 23:52)	548881.974	0.000	0.000
1396335180 (14/03/31 23:53)	548881.974	0.000	0.000
1396335240 (14/03/31 23:54)	548881.974	0.000	0.000
1396335300 (14/03/31 23:55)	548881.974	0.000	0.000
1396335360 (14/03/31 23:56)	548881.974	0.000	0.000
1396335420 (14/03/31 23:57)	548881.974	0.000	0.000
1396335480 (14/03/31 23:58)	548881.974	0.000	0.000
1396335540 (14/03/31 23:59)	548881.974	0.000	0.000

Table 3.1: An excerpt of the water readings in AMPds.

3.1.4 Other Datasets

The only other published dataset that has both electricity and water information is iAWE [1]. Unfortunately, the water meter is only set to pulse every 10 L. This is too low of granularity to produce any meaningful results, especially when water-use actions can be as low as a few litres (as in the dishwasher wash/rinse actions in AMPds). Additionally, the dataset has many false positives (far outnumbering the amount of true positives) with no clear method of cleaning without more insight into the hardware.

3.2 Extracting Activities

Extracting instances of activities from the time series should be as simple as cutting out segments with nonzero electricity readings. The actions of the machine are governed by a timer which must be consuming electricity throughout the entire activity. We assume that the electricity and water patterns of water-use actions are consistent enough between different activity types, even with slightly variable fill rates. Since the current reading is only accurate to the nearest deciampere, the appliance appears to shut off in between actions. Power turns out to be a better indicator than current as to whether the appliance is on or off at any given point in time, since it is accurate to the nearest watt (Figure 3.1).

This is only applicable when looking at the submetered readings. Subtleties such as this will not be found by the disaggregator. Since current is used for disaggregation, instances where only the timer is running will be lumped together with instances where the device is completely off. The peak power reading for this state will be 0 watts.

3.2.1 Dishwasher Examples

Looking at the extracted dishwasher examples by hand, two main patterns arise. These two activities often occur one after the other. Using this information, it was found from the homeowner that although the heavy dishwasher cycle (Figure 3.1a) was the only setting used, sometimes a hold cycle (Figure 3.1b) would be performed if the soap did not dissolve completely. Sometimes these two events happen so close together that they end up being extracted as one (Figure 3.1c). The logarithmic scale in these diagrams shows that if current were used to segment the time series, only single electricity-use actions (one activation of the water pump for a wash or rinse) would be extracted instead of entire activities.

Focusing on the heavy dishwasher cycle, the last segment with low electricity consumption (which shows up as 0 amperes) corresponds to a drying period. Since the heated dry option is not selected, only the timer is running here. Sometimes the machine is prematurely interrupted, affecting the length of this action. These can safely be ignored since they do not correspond to water-use. For all future dishwasher examples, we focus on the first 120 minutes of heavy wash cycles.

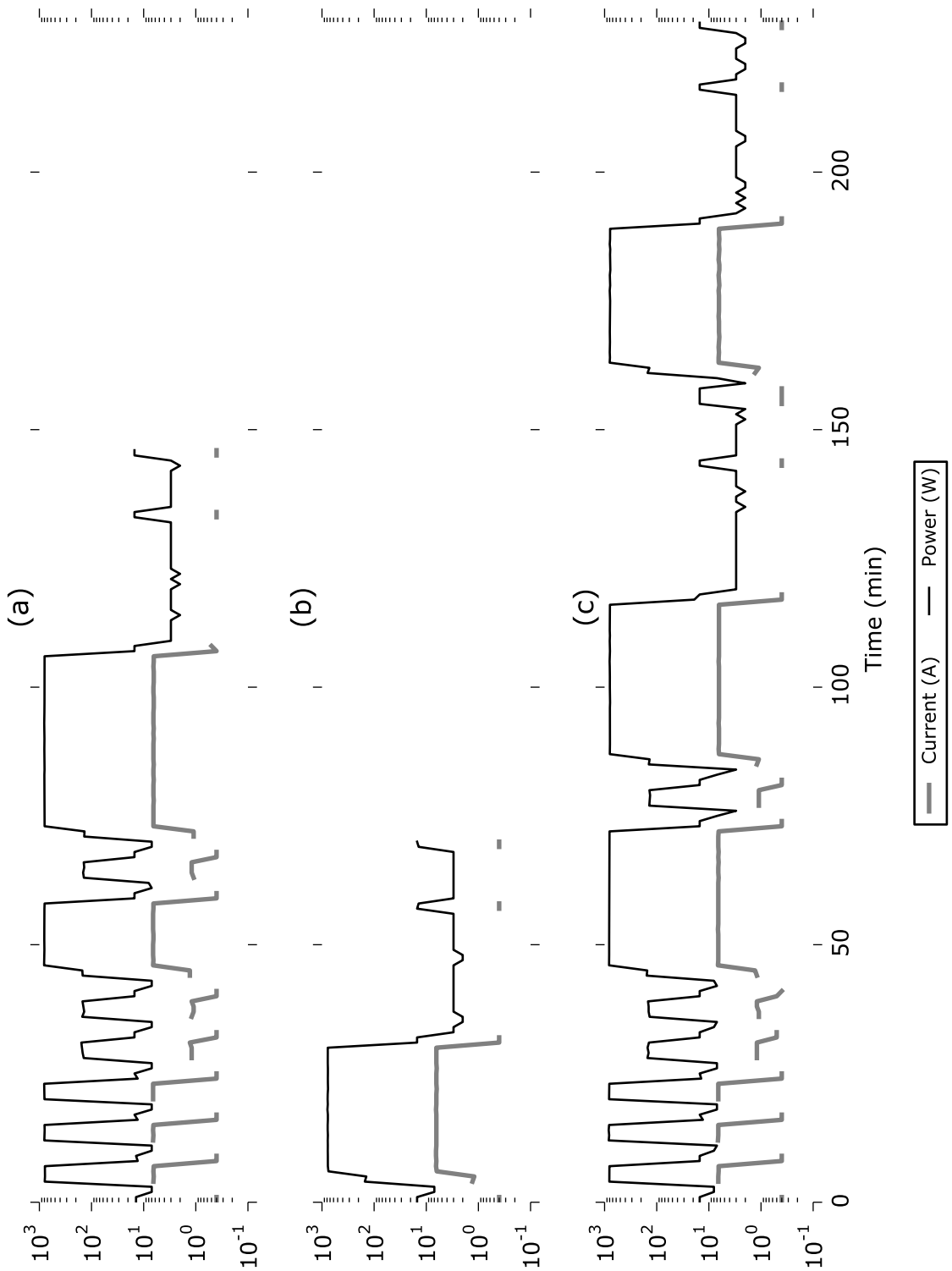

Figure 3.1: Power is the best indicator of where to segment the time series into activities. Looking at the current, the dishwasher appears to shut off in the middle of a run due to the lower precision of the readings. The homeowner always used the heavy wash setting (a). This was followed by a hold if the soap did not break up completely (b). Sometimes a heavy wash and a hold are extracted as one (c).

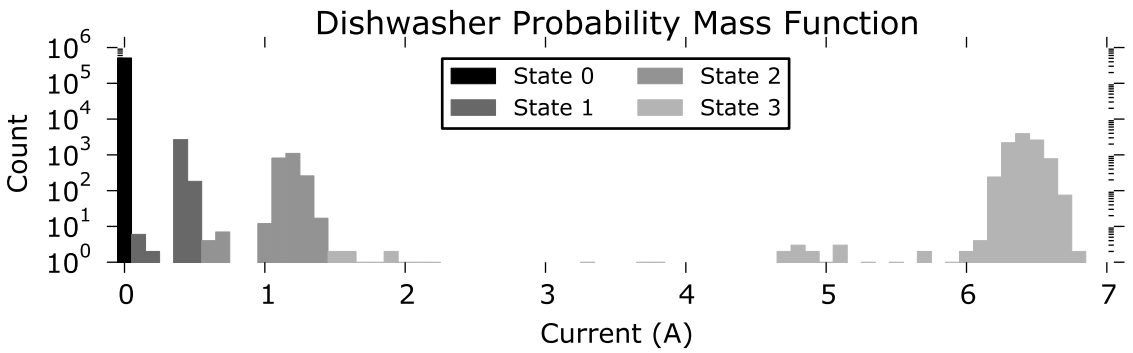

Figure 3.2: Dishwasher probability mass function. Most of the time the dishwasher is in the OFF state. When it is ON, it is in one of three other distinct states.

Figure 3.2 shows the probability mass function for the dishwasher's current reading. Note that the logarithmic scale means that readings of 0 far outnumber other readings. This is the OFF state. When the machine is ON, three additional distinct peaks form. Using the method from [32], the diagram has been separated into states based on these peaks which are used to discretize the dishwasher's raw current reading. A peak is identified when the slope on the left is positive and the slope on the right is negative. Peaks that do not exceed a threshold ϵ (tuned for the dataset) are ignored as noise.

These can be seen applied to three dishwasher examples in Figure 3.3. It is easy to see how small variations in readings can be grouped together into one of four distinct states. The key point here is that the exact current reading does not matter, but rather the state the machine is in. Figure 3.4 shows the whole house water readings during these three examples. Looking at the state changes alongside, a clear pattern emerges.

The areas of water consumption due to the dishwasher are circled. Note that they always correspond to a state change, but the water peak is at the end of a period in the OFF state (State 0). This is due to slight differences in timing for the recording of the two meters. What is more, the sum of the water consumption at any one of these peaks and the following point in time (and sometimes the preceding point in time) always add up to 3 L (minus any noise from other water-use). This explains why the peaks are not always the same. A higher peak is always followed by a slightly lower reading.

Compare the water consumption (integral of the flow rate) from around 20 minutes to 40 minutes in the three examples. The peaks in Figure 3.4a (2.5 L) are higher than the peaks in Figure 3.4c (2 L), but the following readings in Figure 3.4a (0.5 L) are lower than the following readings in Figure 3.4c (1 L). Both add up to 3 L ($2.5\text{L} + 0.5\text{L} = 3\text{L}$ and $2\text{L} + 1\text{L} = 3\text{L}$). In Figure 3.4b the plateaus also add up to 3 L ($1.5\text{L} + 1.5\text{L} = 3\text{L}$). The coarse granularity of readings causes the exact timing of the start of the run to affect the recorded pattern. A reading of 3 L/min is never recorded because it takes somewhere between 1 and 2 minutes for each of these 3 L spurts of water.

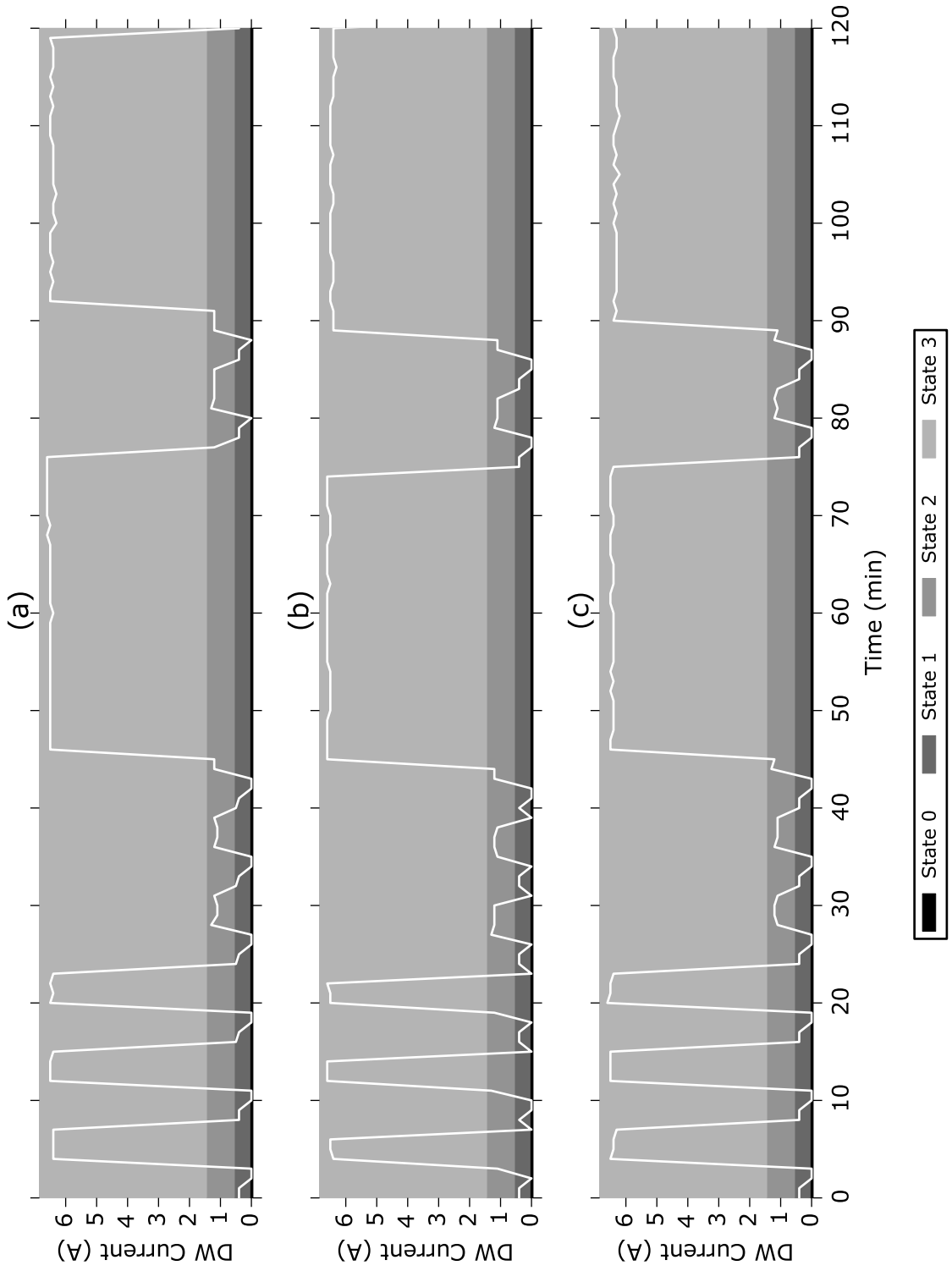

Figure 3.3: Current readings for three dishwasher examples. States emerge by observing certain readings that the machine tends towards. The exact current reading does not matter, but rather the state the machine is in. All three examples follow a similar pattern of state transitions. There are variations between samples due to timing differences.

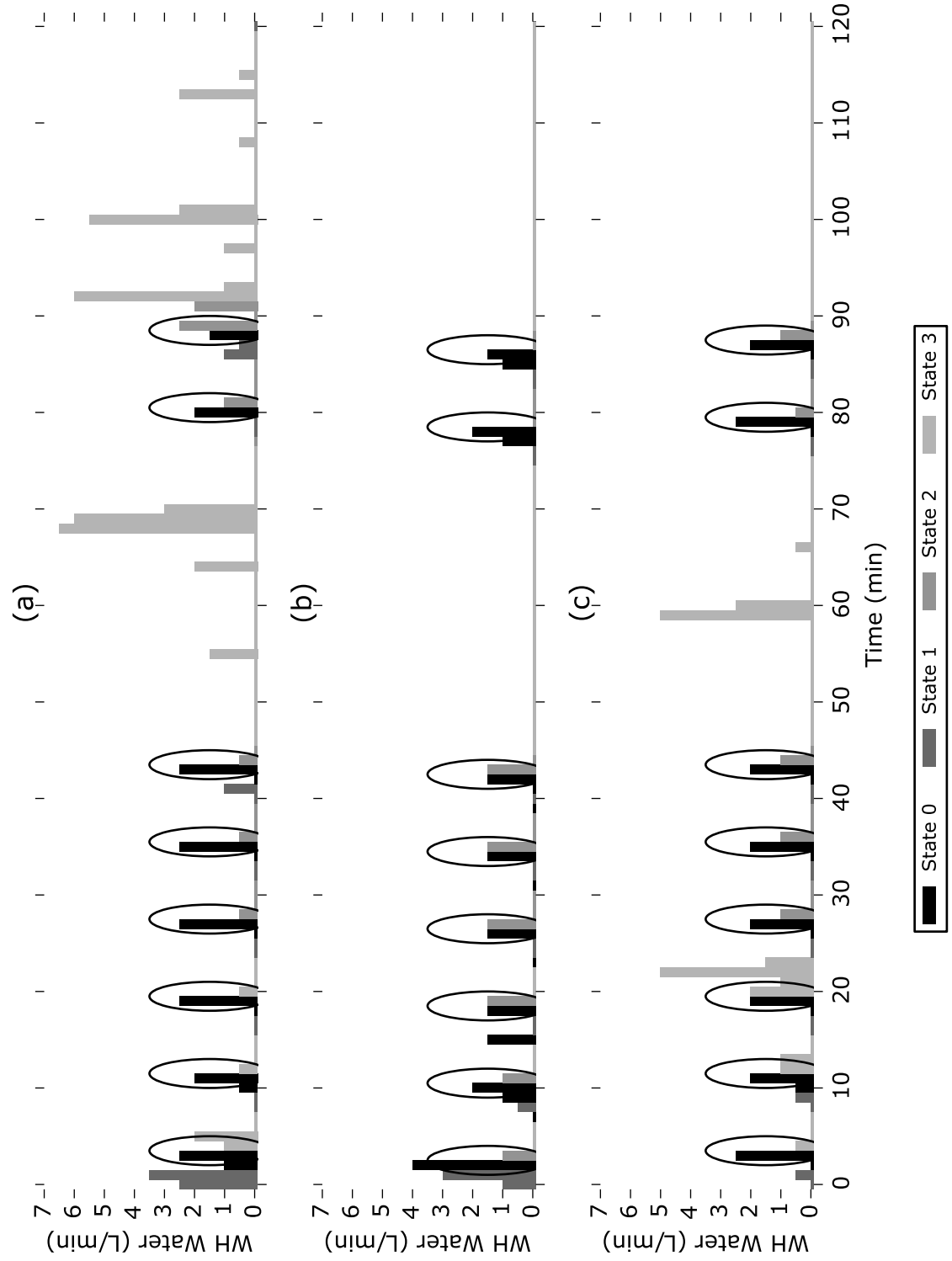

Figure 3.4: Whole house water readings for three dishwasher examples. There are variations between samples due to timing differences. Within a sample, a repeating pattern can be seen. For example, readings of 2.5 L/min followed by 0.5 L/min (a), readings of 1.5 L/min followed by 1.5 L/min (b), or readings of 2 L/min followed by 1 L/min (c). Other readings are noise from the rest of the house.

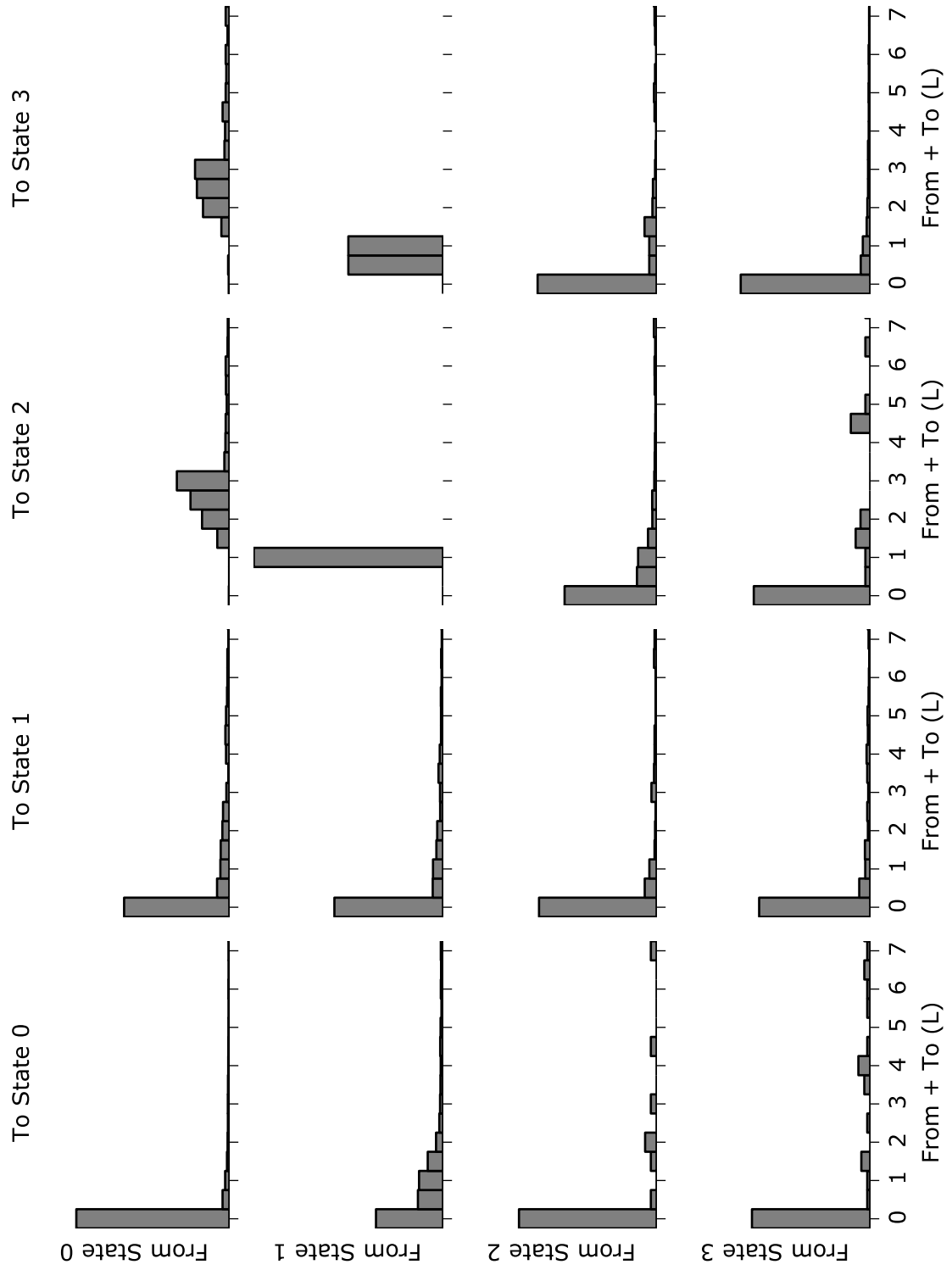

Figure 3.5: Probability mass functions for the sum of water readings at two points in time, conditioned on the electricity state being transitioned *from* and the electricity state being transitioned *to*. Note that when transitioning from State 0 to State 2 or State 3 the peak is around 3 L. Otherwise the peak is at 0 L (transitioning from State 1 to State 2 or State 3 being outliers).

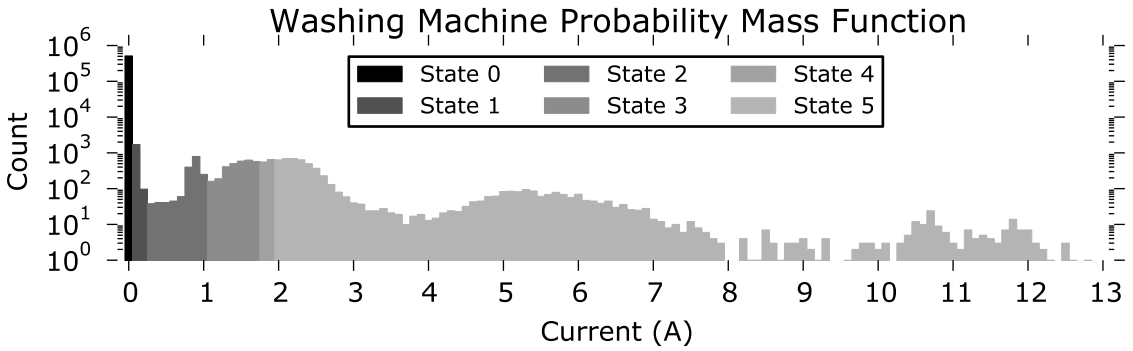

Figure 3.6: Washing machine probability mass function. Most of the time the washing machine is in the OFF state. When it is ON, it is in one of five other distinct states. Notice how close together States 1 through 4 are and the large variability in State 5.

Given this observation, we generated a series of probability mass functions for the sum of water readings at two points in time, conditioned on the transition between electricity states (Figure 3.5). As expected, the peaks in State 0 to State 2 and State 0 to State 3 are around 3 L. Otherwise the peak is usually at 0 L. State 1 to State 2 and State 1 to State 3 are outliers due to low frequency of occurrence.

3.2.2 Washing Machine Examples

Looking at the washing machine data, it does not appear to follow as simple of a pattern as the dishwasher. Figure 3.6 shows the probability mass function for the washing machine's current reading. The method from [32] does not produce as clear of states as in the case of the dishwasher. States 1 through 4 are very close together, so it is hard to tell where the exact boundaries are. State 5 encompasses a large range of readings that vary quite a bit in frequency.

Figure 3.7 and Figure 3.8 show three examples of washing machine runs. There does appear to be some sort of pattern in the current readings in Figure 3.7, but the state transitions are not as clear as in the case of the dishwasher. As with the dishwasher, this machine is always set to the heavy cycle. The manual states: "This cycle provides 16 minutes of reversing tumble wash action for heavily soiled regular items, followed by 3 rinses plus an automatic extra rinse and a final spin." [18] By the sounds of this, the drum should be filled up five times (circled on Figure 3.8, as with the dishwasher).

These examples suggest that it takes more than 2 minutes to fill up the drum. A rough estimate is that it has to fill about 17 L over three or four data points. Here there are six electricity states, which would lead to 36 possible transitions. Taking three data points into account instead of two, there would be 216 PMFs. Clearly this will quickly get out of hand and will not be reliable since the states are so close together.

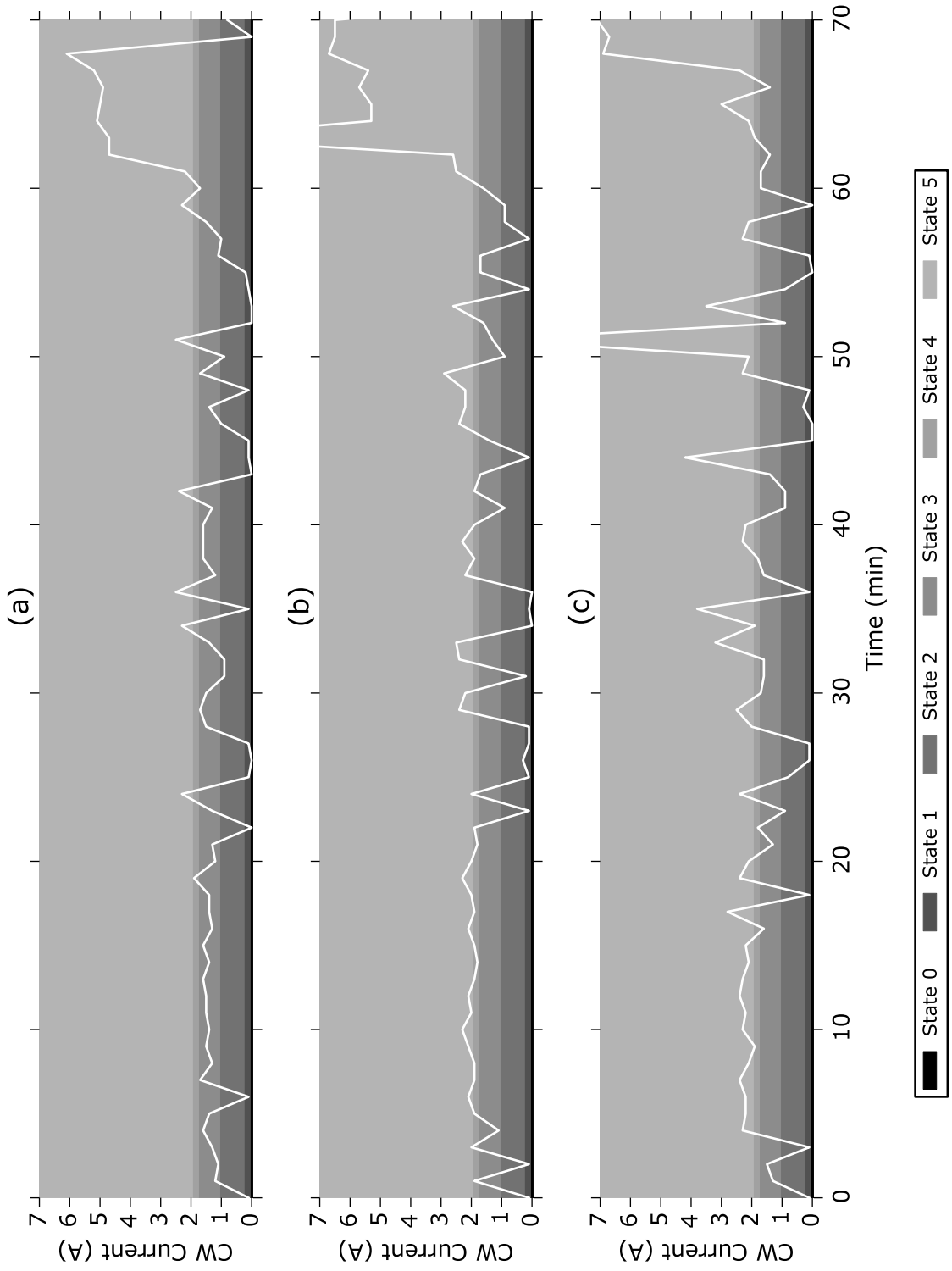

Figure 3.7: Current readings for three washing machine examples. There does appear to be some sort of pattern, but it is not as distinct as in the case of the dishwasher. Many of the electricity states are close together and hard to distinguish. The figure is truncated at 7 A to make it easier to see some of the smaller states. Note the sporadic readings in State 5.

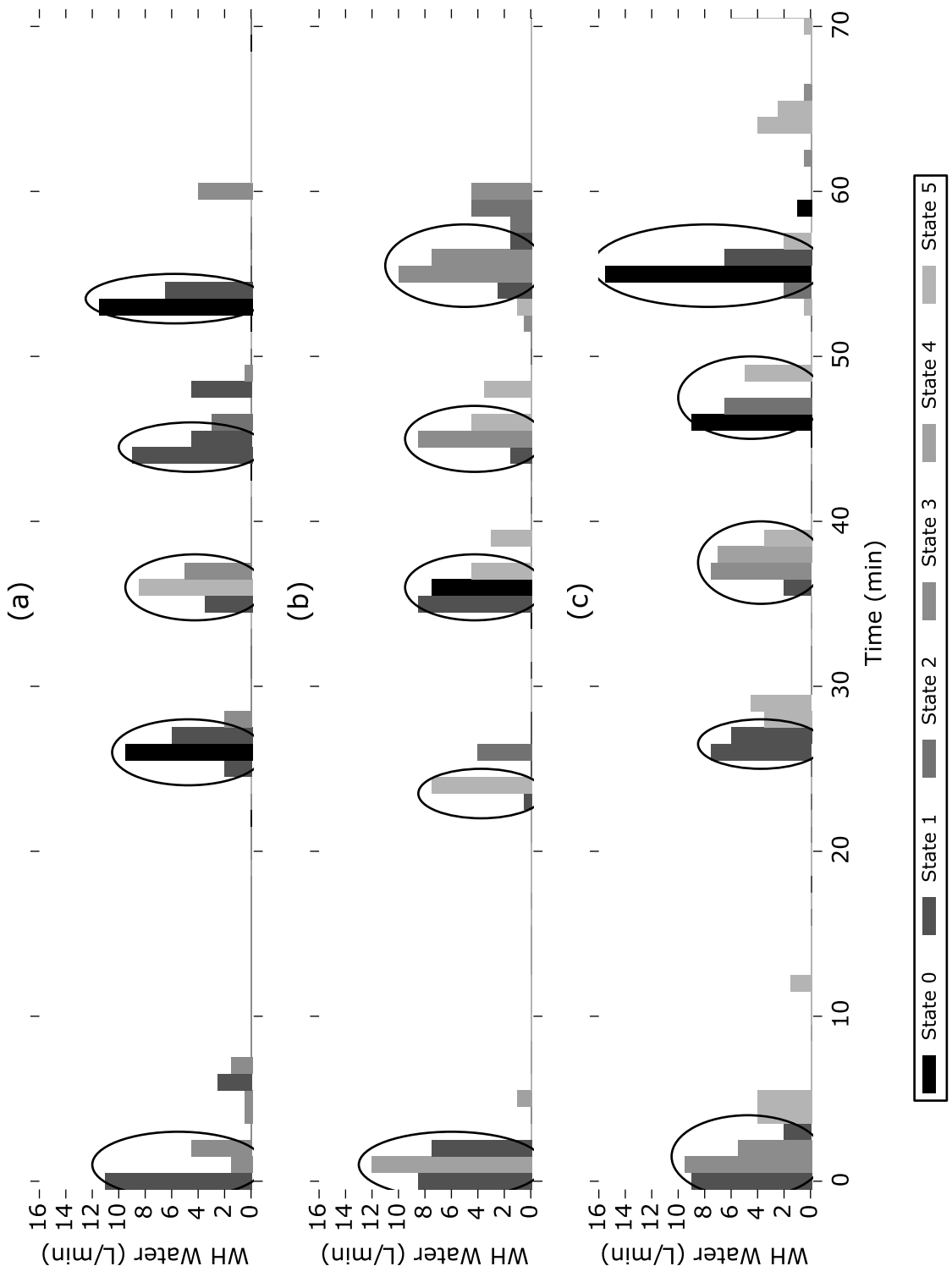

Figure 3.8: Whole house water readings for three washing machine examples. Five water-use actions can be seen in each activity, but it is hard to determine how much is due to the washing machine and how much is noise from the rest of the house. Each action is weakly correlated with electricity State 1. In the fourth action in example (c) this is the 0 L/min water reading in the middle.

3.3 Learning Patterns

We only have submetered data for the electricity loads, but the result we are trying to learn is an approximation of the unmetered water loads. Originally we had planned to exploit the fact that machines follow set consumption patterns. These patterns proved too cumbersome to learn, as slight differences in timing and unpredictable environmental factors compound, resulting in a wide range of variability within activities of the same type (§3.2).

It turns out that only the local context around each action is necessary to provide a good indication of the machine's water consumption. Hidden Markov Models (HMMs) are designed to capture these kinds of relationships. Our solution requires some modifications to the traditional formulation. Like an unsupervised scenario, we do not have labelled data to train with. Unlike an unsupervised scenario, we have some prior knowledge about the output we are looking for (i.e. each appliance's water-use is bound above by the whole house's water reading). Here we present the Capped Viterbi Algorithm for applying a supervised method to an unsupervised disaggregation problem.

The distinction between activity and action is still important when extracting samples for training and labelling. As described in §3.2, water-use actions that correspond to electricity-use actions tend to line up where the machine is in the OFF electricity state. Entire activities must be extracted to find these relationships between actions.

3.3.1 Review of Hidden Markov Models

HMMs provide a machine learning method for efficiently modelling and learning patterns in state transitions over time [36, 22]. They rely on the Markov assumption that a state at a given point in time being dependent on all previous states can be simplified to only being a conditional probability on a single previous state. A standard HMM can be defined as

$$\text{HMM} = \{h, o, \mathbf{S}, \mathbf{T}, \mathbf{E}\},$$

where h is the number of possible hidden states, o is the number of possible observed states, $\mathbf{S}$ is the start probability vector, $\mathbf{T}$ is the transition probability matrix, and $\mathbf{E}$ is the emission probability matrix.

An HMM describes a relationship between a finite sequence of hidden states and a finite sequence of observed states

$$H = (h_0, h_1, h_2, \ldots, h_T), h_t \in 0..h-1$$
$$O = (o_0, o_1, o_2, \ldots, o_T), o_t \in 0..o-1.$$

In a first-order HMM, the probability of being in a given hidden state at time t is only conditioned on the hidden state at time $t-1$. The probability of seeing a given observed state at time t is only conditioned on the hidden state at that time. $\mathbf{S}$ is an h element

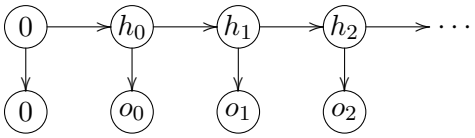

Figure 3.9: A graphical representation of a first-order HMM with 0 start states.

vector, $\mathbf{T}$ is an $h \times h$ matrix, and $\mathbf{E}$ is an $h \times o$ matrix where

$$s_j = P(h_0 = j)$$
$$t_{i,j} = P(h_t = j | h_{t-1} = i)$$
$$e_{j,n} = P(o_t = n | h_t = j).$$

In the problem we are trying to solve, we know the appliance is in the OFF state (State 0) outside of the extracted samples regardless of the representation we choose. We prepend a 0 hidden state to H and a 0 observed state to O to simplify the start probabilities. With this $\mathbf{S} = [1]$, a simple one-element vector since the probability of starting in State 0 is 100%. Figure 3.9 shows a graphical representation of such an HMM. This simplifies things when we look at higher-order models later.

The Viterbi Algorithm is a dynamic programming solution to the problem of finding the most likely sequence of hidden states corresponding to a sequence of observed states [43, 17]. Figure 3.10 shows the pseudocode. Running through the steps of the Viterbi Algorithm can be visualized with a trellis diagram (Figure 3.11). At each point in time, only the most likely path leading there needs to be considered.

3.3.2 Capped Viterbi

For our particular problem, we know the electrical state of the machine and the whole house water reading at each point in time. We want to find the disaggregated water reading for the machine in question. Since we do not have submetered water data, we cannot use a standard supervised learning method. Conversely, we do not want to use a standard unsupervised learning method, because we have prior knowledge about the disaggregated water reading. That is, we know that it must be in half-litre increments and is bound above by the whole house water reading.

To utilize this information, we formulate the model as we would if we were learning the whole house water reading given only the electrical state of one machine. This means the hidden states are the range of possible whole house water readings in half-litre increments and the observed states are the machine's electrical states. Since we already have both of these pieces of information, we can use a supervised training method. The intuition behind this is that we are purposely 'under-fitting' the data in the hopes that we are left with only

Input: h
$O = (0, o_0, o_1, o_2, \ldots, o_T)$
$\mathbf{S} = [s_j]$
$\mathbf{T} = [t_{i,j}]$
$\mathbf{E} = [e_{j,n}]$
Output: $H = (h_0, h_1, h_2, \ldots, h_T)$
for $j \in S$ **do**
  $H[j] \leftarrow ()$; `// stores the most likely sequence` H `ending in` j
  $P[j] \leftarrow s_j \cdot e_{j,0}$; `// shows the probability of` $H[j]$
end
for $t \leftarrow 0$ **to** T **do**
  for $j \leftarrow 0$ **to** $h - 1$ **do**
    $newP[j] \leftarrow \max_{i \in P}(P[i] \cdot t_{i,j} \cdot e_{j,o_t})$;
    $newH[j] \leftarrow \text{concatenate}(H[\arg\max_{i \in P}(P[i] \cdot t_{i,j} \cdot e_{j,o_t})], j)$;
  end
  $P \leftarrow newP$;
  $H \leftarrow newH$;
end
return $H[\arg\max_{i \in P}(P[i])]$

Figure 3.10: The Viterbi Algorithm for first-order HMMs.

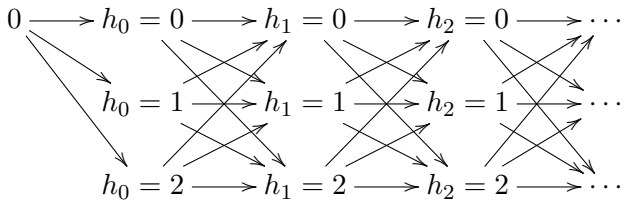

Figure 3.11: The steps of the Viterbi Algorithm can be viewed as a trellis.

the disaggregated reading we are looking for when predicting. Instances where the machine in question is the only thing consuming water (i.e. the whole house water reading is exactly the desired answer) are more consistent with each other and train the model to ignore noise from the rest of the house.

Not only do we have access to the true hidden labels at training time, but we also know them for the sequences we want to predict. Of course we do not want our model to outright see this information, as this would just leave us with the whole house water readings we already know. Instead, we provide this information as a 'hint' to the Viterbi Algorithm.

We call this new variant the Capped Viterbi Algorithm. In addition to the sequence of observed states O, our algorithm takes as input a sequence of upper bounds (or 'caps') on the hidden states,

$$C = (c_0, c_1, c_2, \ldots, c_T), c_t \in 0..h - 1,$$

Input: $C = (c_0, c_1, c_2, \ldots, c_T)$
$O = (0, o_0, o_1, o_2, \ldots, o_T)$
$\mathbf{S} = [s_j]$
$\mathbf{T} = [t_{i,j}]$
$\mathbf{E} = [e_{j,n}]$
Output: $H = (h_0, h_1, h_2, \ldots, h_T)$
for $j \in S$ **do**
  $H[j] \leftarrow ()$; `// stores the most likely sequence` H `ending in` j
  $P[j] \leftarrow s_j \cdot e_{j,0}$; `// shows the probability of` $H[j]$
end
for $t \leftarrow 0$ **to** T **do**
  for $j \leftarrow 0$ **to** c_t **do**
    $newP[j] \leftarrow \max_{i \in P}(P[i] \cdot t_{i,j} \cdot e_{j,o_t})$;
    $newH[j] \leftarrow \text{concatenate}(H[\arg\max_{i \in P}(P[i] \cdot t_{i,j} \cdot e_{j,o_t})], j)$;
  end
  $P \leftarrow newP$;
  $H \leftarrow newH$;
end
return $H[\arg\max_{i \in P}(P[i])]$

Figure 3.12: The Capped Viterbi Algorithm for first-order HMMs.

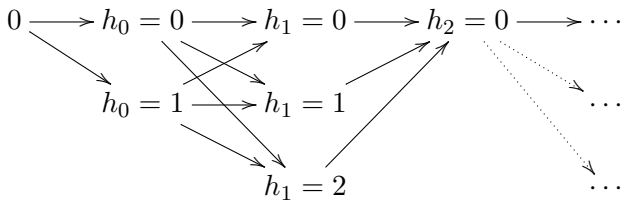

Figure 3.13: Reduced trellis for Capped Viterbi Algorithm (here $C = 1, 2, 0, \ldots$).

to reduce the search space. Here, this is the whole house water-use (used as H when training). Figure 3.12 shows the pseudocode. Figure 3.13 exemplifies a reduced trellis.

3.3.3 Our Model

A simple first-order model is not able to capture the relationships in our dataset. Given our observations in §3.2, we tried building a model that correlates the sum of water readings at two points in time with a pair of electrical states. This results in $h + h = 2h$ hidden states and $o \cdot o = o^2$ observed states. Figure 3.14 shows a graphical representation of this model. $\mathbf{T}$ is a $2h \times 2h$ matrix and $\mathbf{E}$ is a $2h \times o^2$ matrix where

$$t_{i,j} = P(h_{t-1} + h_t = j | h_{t-2} + h_{t-1} = i)$$
$$e_{j,(m,n)} = P(o_{t-1} = m, o_t = n | h_{t-1} + h_t = j).$$

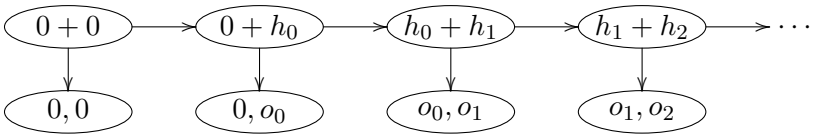

Figure 3.14: A first-order HMM where pairs of points in time are taken together. The sum of two hidden states is correlated with a tuple of observed states.

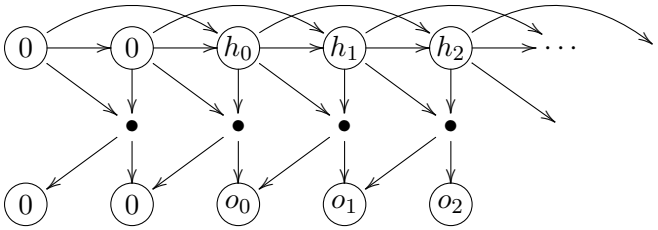

Figure 3.15: When the pairs of hidden states in Figure 3.14 are taken as a tuple instead of a sum, the result is similar to a second-order HMM. The relationship with the observed states does not follow that of a standard second-order HMM. Bullets line up with the points in time that emission pairs are taken into account when predicting the optimal sequence of hidden states.

Unfortunately, this did not yield good results. Had this method worked, there still would have been the problem of separating the summed pairs of water readings into individual points in time. One of the goals of disaggregation is to not have to perform filtering or resampling.

As it turns out, we found taking tuples of hidden states into account (as with the observed states) instead of sums gives better predictions. This has the added benefit of inherently representing individual points in time. We can simplify Figure 3.14 to a second-order model with $\mathbf{T}$ being a $h^2 \times h$ matrix and $\mathbf{E}$ being an $h^2 \times o^2$ matrix where

$$t_{(i_2,i),j} = P(h_t = j | h_{t-2} = i_2, h_{t-1} = i)$$
$$e_{(i,j),(m,n)} = P(o_{t-1} = m, o_t = n | h_{t-1} = i, h_t = j).$$

Figure 3.15 shows the modified model.

It is important to note that this emission matrix violates the generative property of HMMs. Observations are generated in overlapping pairs, with no matching constraints. In a generative sense, this is of course impossible, since each point in time may only have one true observed state. Since we are only using the model in a discriminative fashion, this is okay. The only difference is over which terms the matrix is normalized to make its entries valid probabilities. Compare this to a standard second-order model where the emission matrix is the same as in a standard first-order model. This pattern is easily generalized to higher-order models. In Chapter 4 we provide an evaluation of different ordered models.

3.3.4 Time Complexity

There are h^T possible sequences of hidden states for a given sequence of observed states. The standard first-order Viterbi algorithm finds the optimal sequence by only looking at h possibilities for each hidden state at each point in time. This results in an $O(h^2T)$ running time. Generalizing to order-n models, it looks at h^n possibilities for each hidden state at each point in time, giving a running time of $O(h^{n+1}T)$. Note that the growth is exponential in terms of the order of the model, but is a fixed polynomial in terms of the number of hidden states for a particular model.

Capped Viterbi Algorithm reduces the search space by only looking at $\prod_{x=t-n}^{t-1} c_x \leq h^n$ possibilities for each hidden state up to $c_t \leq h$ at each point in time. In the worst case $C = (h, h, \ldots, h)$, keeping the $O(h^{n+1}T)$ running time. Empirically the running time is much less than this, as usually little to no water is being used in the house relative to the highest recorded water consumption (i.e. $c_t \ll h$). Running time is also kept low by only keeping track of subsequences with non-zero probabilities. This is especially significant in higher-order models where $\mathbf{T}$ is quite sparse.

3.3.5 Smoothing

To ensure there is at least one possible path with a non-zero probability at each point in time, we make every entry in the 0 column of $\mathbf{T}$ non-zero (i.e. every row can transition to OFF). This maintains the sparsity of $\mathbf{T}$ while allowing any path to 'zero out'.

For the emission matrix, we are not concerned with sparsity. The observed states are fixed by the input and the possible hidden states are kept low by the sparse transition matrix. Therefore for every subsequence of hidden states encountered in the training data, we can safely smooth over all possible combinations of observed states to ensure there are no zero counts. The simplest form of smoothing is Laplace smoothing. Here 1 (or some $\delta < 1$) is added to every count. In practice, this does not perform well [8].

For our purposes, $\mathbf{E}$ is smoothed by averaging out the 0 and 1 counts in each row before normalizing. Effectively, the singular counts are spread out to emissions with no count. The intuition behind this is that the number of fluke single observations give an indication of the probability of a previously unseen emission, similar to Good-Turing smoothing [24]. In cases where there are no 1 counts, all of the 0 counts are set to 1.

3.4 Summary

Here, we formalized the task at hand with a concrete problem definition and introduced the dataset we used. We provided a method for extracting activities from the dataset and show the intuition behind how the water data can be correlated with the electricity data. Finally, we described our application of HMMs to disaggregate household water readings.

Chapter 4

Experimental Results

In this chapter, we provide the results of a formal evaluation of the performance of our model on AMPds. By hand-labelling the dishwasher water data, we are able to conduct a quantitative analysis. For the washing machine a qualitative analysis is conducted, as the patterns in the data are too inconsistent to reliably hand-label.

4.1 Dishwasher

Using the method outlined in §3.2, 185 dishwasher activities were extracted from the second year of AMPds. These were divided into 10 sets with 18 or 19 activities each. 10-fold cross-validation was used to evaluate the performance of first-order, second-order, third-order, and forth-order models of the variety described in §3.3. The results are presented in Table 4.1 along with average running times per data point.

The first-order model is not able to capture any of the relationships in the dataset, as shown by the explained variance of 0. In fact, it just labels the dishwasher's water consumption as 0 L/min at every point in time. This acts as a good baseline because it shows that even only looking at times when the dishwasher is running, most of the time it is not consuming water. A mean squared error of under 0.25 L^2/min^2 is trivial to achieve.

Order	Explained Variance	Mean Squared Error (L^2/min^2)
—	-33.648582	9.238157
1	0.000000	0.249550
2	0.932241	0.015590
3	**0.938049**	**0.014253**
4	0.902036	0.022544

Table 4.1: Results of 10-fold cross-validation on different orders of models. The first row shows the results if we do not perform disaggregation and just assume the whole house water meter reading is entirely due to the dishwasher.

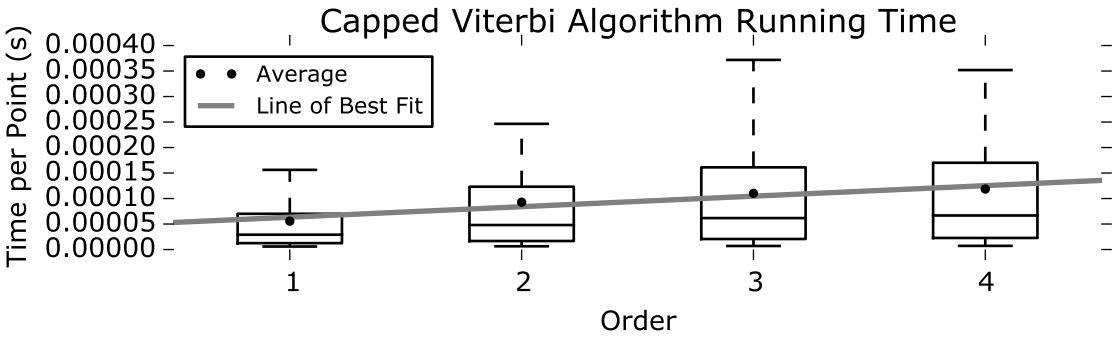

Figure 4.1: Box plots showing the running times of the Capped Viterbi Algorithm for the experiments in Table 4.1. Times are normalized by the number of points in the data series being labelled.

As expected, the second-order model performs considerably better, with a mean squared error of 0.015590 L^2/min^2. The third-order model shows minor improvements over this, achieving a mean squared error of 0.014253 L^2/min^2. Once we reach the fourth-order model, we begin to see diminishing returns, with a mean squared error of 0.022544 L^2/min^2. This is due to the extreme sparsity of such a high order model.

Even though we are using discrete models, we chose to use continuous metrics for evaluation. This is because the hidden states we are predicting (water-use) are from a continuous space that has been discretized by the nature of a pulse-based meter. Discrete metrics, such as F-measure, do not take the orderability of such a set into account. A prediction is either correct or incorrect, because there is no notion of closeness. By using continuous metrics we are able to provide more meaningful scoring. Predictions that are closer to the truth are penalized less than those that are further. In particular, we focus on the mean squared error due to its popularity, which makes it easier to compare against other works.

In Figure 4.1 we use box plots to display the running times of the Capped Viterbi Algorithm for each order of model. We divide each running time by T, the length of the data series, to focus on the h^{n+1} coefficient in the theoretical upper-bound. The caps, $c_t \leq h$, that contribute to this term are fixed for each example. Since these can vary greatly, we see a large range of running times. Relative to the medians and averages, the interquartile ranges are quite large (58 µs, 107 µs, 140 µs, and 147 µs, for the first-, second-, third-, and fourth-order models, respectively). Despite this, it is easy to see that the running time does not grow exponentially in terms of the order of the model and we are able to use a linear best fit. Looking at the time to label an entire dishwasher run, even for the fourth-order model it takes at most 150 ms in the most extreme case seen in AMPds.

Figure 4.2 shows the hand-labelled ground truth for the three examples from Figure 3.3 and Figure 3.4. The output of the third-order model when run on these examples is shown in Figure 4.3. We are able to capture all of the water-use actions almost exactly.

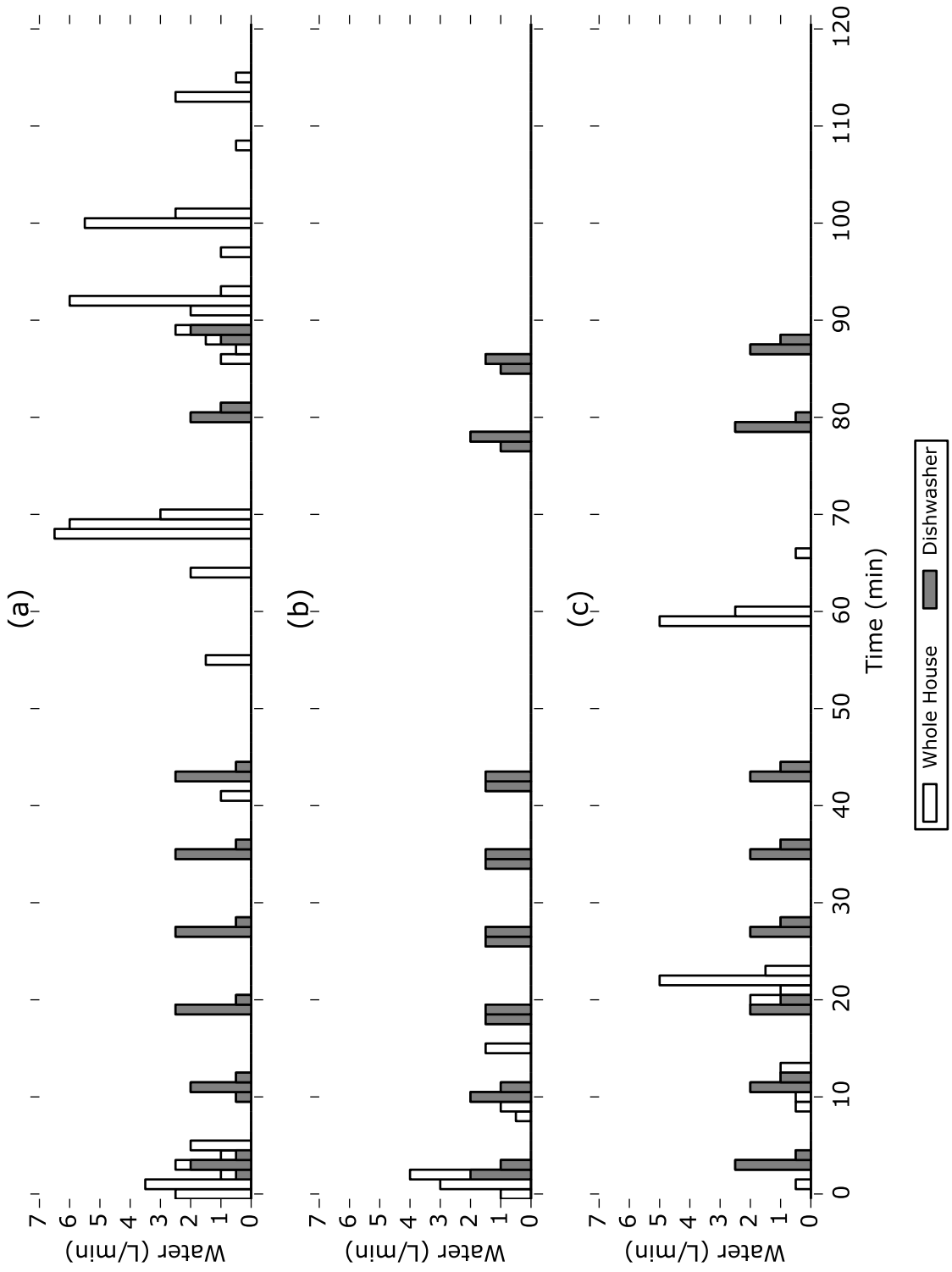

Figure 4.2: Hand-labelled ground truth for three dishwasher examples. The labeller was able to use knowledge from nearby actions to more accurately label difficult cases. For example, there are multiple ways to achieve 3 L in the first action in (a). Looking at the following action allows a human to determine the correct answer.

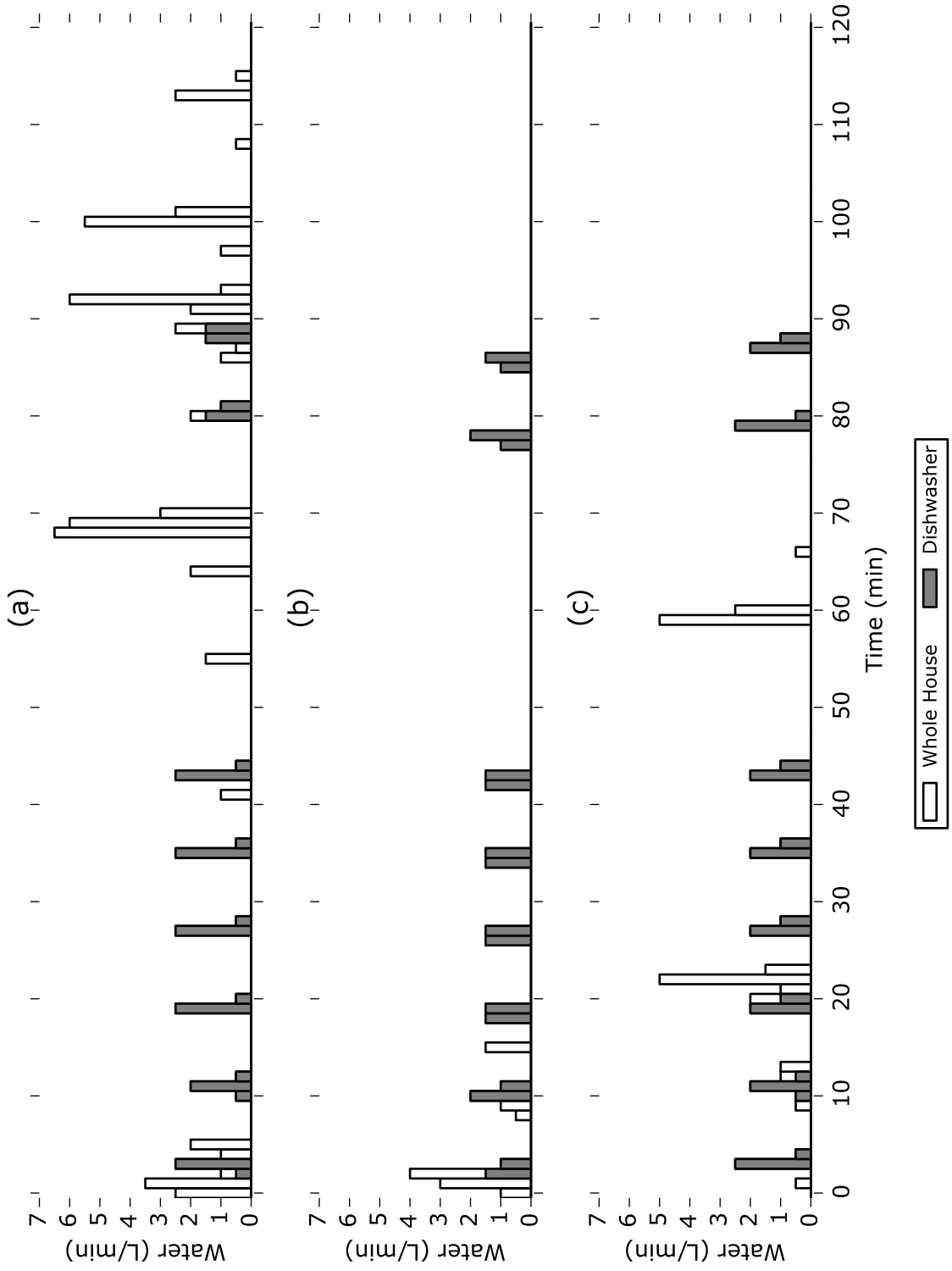

Figure 4.3: Output of third-order model on three dishwasher examples. Compare to the hand-labels in Figure 4.2.

Folds	Explained Variance	Mean Squared Error (L^2/min^2)
1	0.713981	0.062329
2	0.880054	0.026142
3	0.809875	0.041438
4	0.913705	0.018836
5	0.922024	0.017009
6	0.953896	0.010046
7	0.960709	0.008562
8	0.960709	0.008562
9	0.960709	0.008562

Table 4.2: Varying the amount of training data for a third-order model. The tenth fold is fixed as the test set. Results are shown using the first one to nine folds as the training set.

4.1.1 Amount of Training Data

By using 10-fold cross-validation, we are able to utilize all of our limited data for both training and testing purposes. This also means that we use nearly a whole year's worth of data for training for each test. In a real-world setting, this would be a lot of data to require. One of the strengths of our model is that it does not require labelled data to train. This means any additional unlabelled data gathered while using the disaggregator can be used to retrain it to increase accuracy. In other words, the model can improve over time.

We ran tests to see how the amount of training impacted performance. Table 4.2 shows the results of training using an increasing amount of data. The tenth fold is fixed as the test set. As expected, with more training data performance generally improves. With only one fold used for training, the model is still able to reach a mean squared error of 0.062329 L^2/min^2. Two folds worth of data decreases this to 0.026142 L^2/min^2. Introducing the third fold happens to increase it to 0.041438 L^2/min^2. Performance starts to near Table 4.1 with four folds resulting in 0.018836 L^2/min^2. This decline in error bottoms out at 0.008562 L^2/min^2 using seven out of the nine folds.

4.1.2 Gallon Pulses

Additionally, we performed an evaluation on the first few months of AMPds where the meter was set to pulse once every gallon. In this case, the hidden states represent gallons instead of half-litres. The coarser granularity makes it harder to distinguish some of the water-use actions. Since each action uses roughly 3 L of water, some of them are not properly logged, as they do not surpass the gallon (3.785 L) threshold to trigger a pulse.

Using this subset of the data, we extracted 44 dishwasher activities. These were divided into 10 sets with 4 or 5 activities each. As above, 10-fold cross-validation was used to evaluate the performance of first-order, second-order, third-order, and forth-order models. The results are presented in Table 4.3.

Order	Explained variance	Mean squared error (L^2/min^2)
—	-5.715885	2.950385
1	0.000000	0.398634
2	0.000000	0.398634
3	**0.495219**	**0.196856**
4	0.482289	0.201778

Table 4.3: Results of 10-fold cross-validation on the first few months of AMPds with gallon pulses. The first row shows the results if we do not perform disaggregation and just assume the whole house water meter reading is entirely due to the dishwasher.

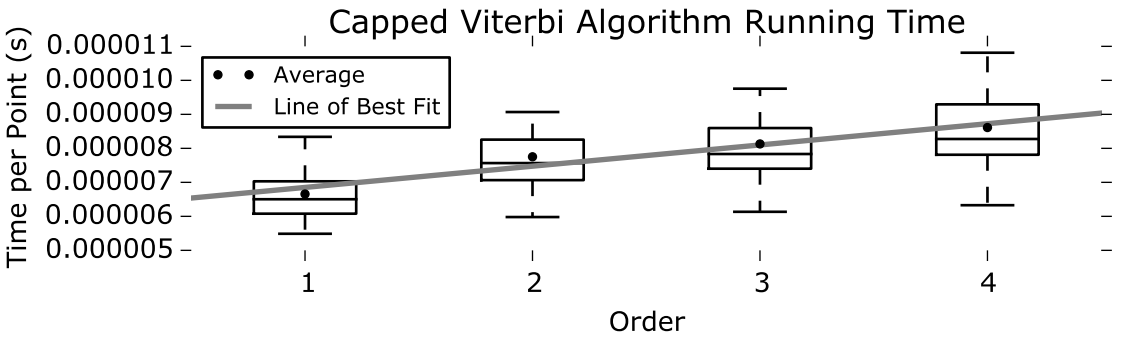

Figure 4.4: Box plots showing the running times of the Capped Viterbi Algorithm for the experiments in Table 4.3. Times are normalized by the number of points in the data series being labelled.

As above, the first-order model always labels with 0 L/min at every point in time. Here the second-order model also fails to capture any of the relationships in the data. Again the third-order model shines, and the fourth-order model sees a decline in accuracy.

The testing times shown in Figure 4.4 exhibit a pattern of growth similar to Figure 4.1. In all cases the running times are much faster because there are fewer hidden states. Here we are met with a simple time–accuracy tradeoff. Not only is the decrease in accuracy too dramatic to warrant using lower granularity readings, the time saved is inconsequential, as the model based on half-litre pulses is able to label an entire dishwasher activity (over 100 data points) in under a second, let alone under the reading rate of once per minute.

Figure 4.5 shows the entire extraction process outlined in §3.2 applied to one of the gallon granularity activities. The dishwasher current PMF for this portion of the dataset is shown in Figure 4.5a. Here there happens to be enough readings around 4 A to form an additional state (c.f. Figure 3.2). This state does not show up in this example (or many activities at all for that matter). These states are applied to our example in Figure 4.5b. States are labelled on the corresponding whole house water readings in Figure 4.5c.

The results of the Capped Viterbi Algorithm on this example are shown in Figure 4.6. Figure 4.6a shows the hand-labelled ground truth. Results of the third- and fourth-order

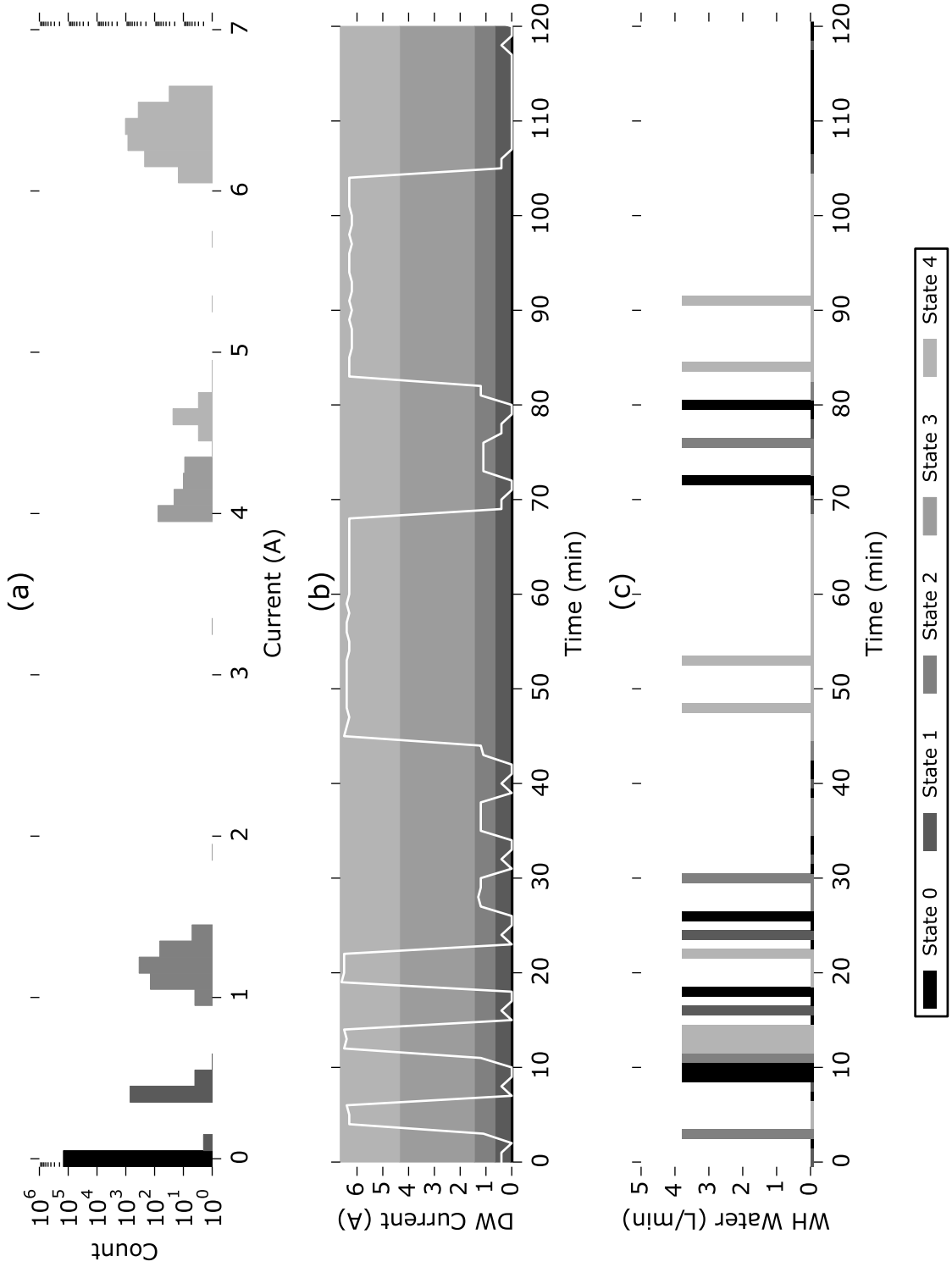

Figure 4.5: The extraction process for a gallon granularity example including the dishwasher current PMF (a), the states applied to the current reading (b), and the corresponding whole house water readings (c).

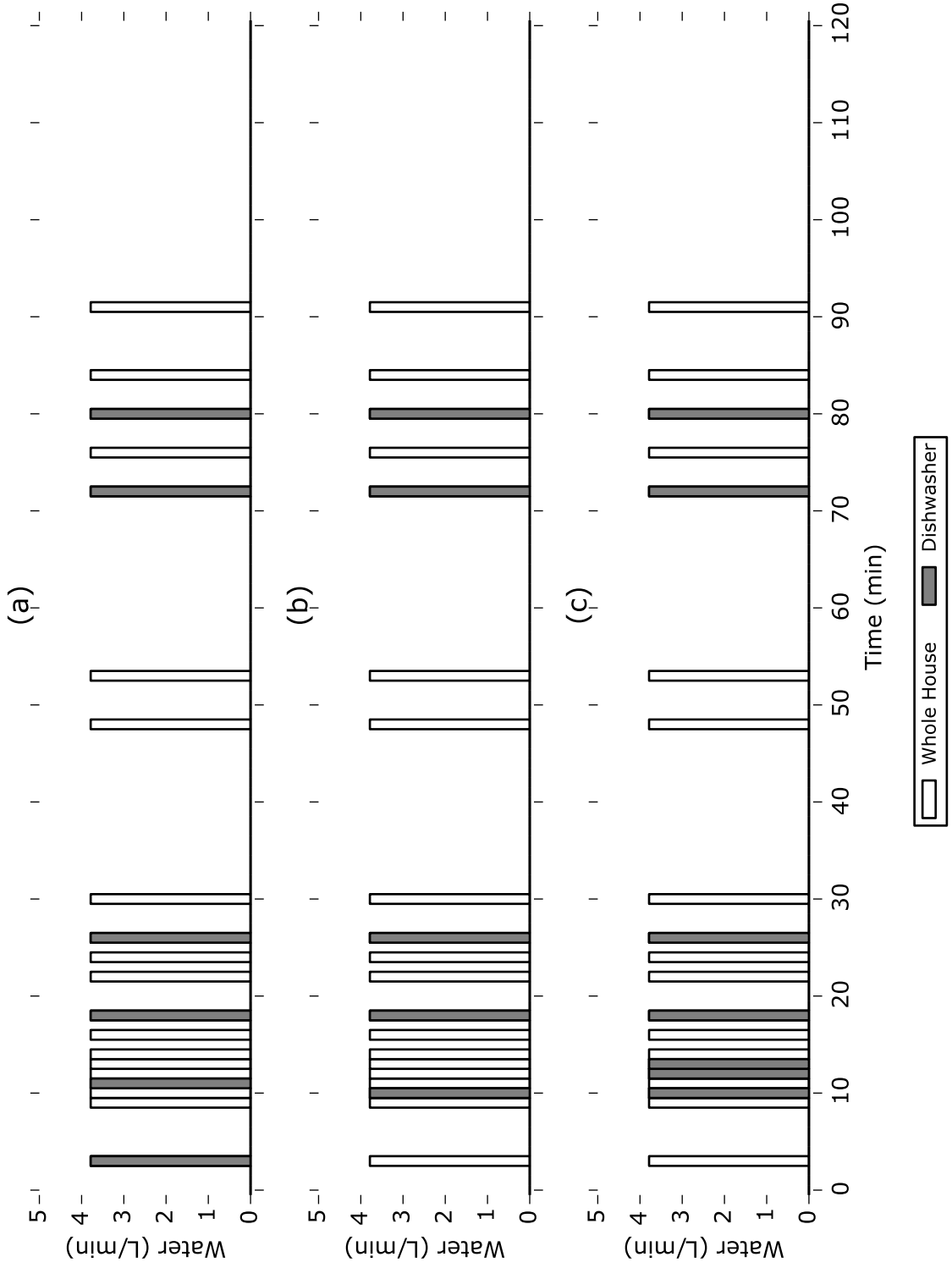

Figure 4.6: Results of running the Capped Viterbi Algorithm on the example from Figure 4.5 including ground truth (a), third-order (b), and fourth-order (c).

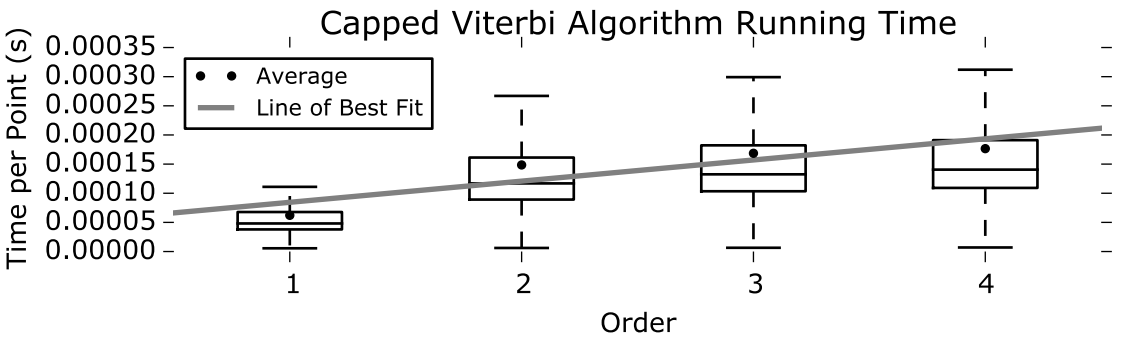

Figure 4.7: Box plots showing the running times of the Capped Viterbi Algorithm for the experiments in Figure 4.8 and Figure 4.9. Times are normalized by the number of points in the data series being labelled.

models are shown in Figure 4.6b and Figure 4.6c, respectively. Referring to Figure 4.5c, we see that the model captures water-use actions that are logged when the machine is in State 0, but not those that appear afterwards in the transition to State 2. This is especially apparent around 10 minutes where both models predict the water-use occurring one minute early. Additionally, the fourth-order model inaccurately labels two extra data points.

4.2 Washing Machine

As stated previously, the readings for the washing machine do not exhibit clear enough patterns to allow for hand-labelling. From the second year of AMPds we extracted 219 washing machine activities. These were split into 10 folds with 21 or 22 activities each. Figure 4.7 shows the running times for this experiment, comparable to Figure 4.1.

Continuing from the examples in Figure 3.7 and Figure 3.8, we perform a qualitative analysis of our model. Consistent with the dishwasher, a third-order model (Figure 4.9) outperforms a second-order model (Figure 4.8). The second-order model is only able to capture two or three out of the five expected water-use actions from Figure 3.8, while the third-order model is able to at least partially capture four or five out of five.

It is difficult to tell how much of the water is noise from the rest of the house. These samples do not appear to be too noisy, so we would expect most of the water consumption around each action to be mostly due to the washing machine. The third-order model still has difficulty with actions where the water reading appears to drop down to 0 L/min in the middle (see the second and fourth actions in Figure 4.9b and the fourth action in Figure 4.9c). Recorded water that is clearly not from the washing machine (see around 10 minutes and 65 minutes in Figure 4.9c) is successfully ignored.

An evaluation on the first few months of AMPds where the meter was accidentally only set to pulse once every gallon was also performed for the washing machine. In contrast

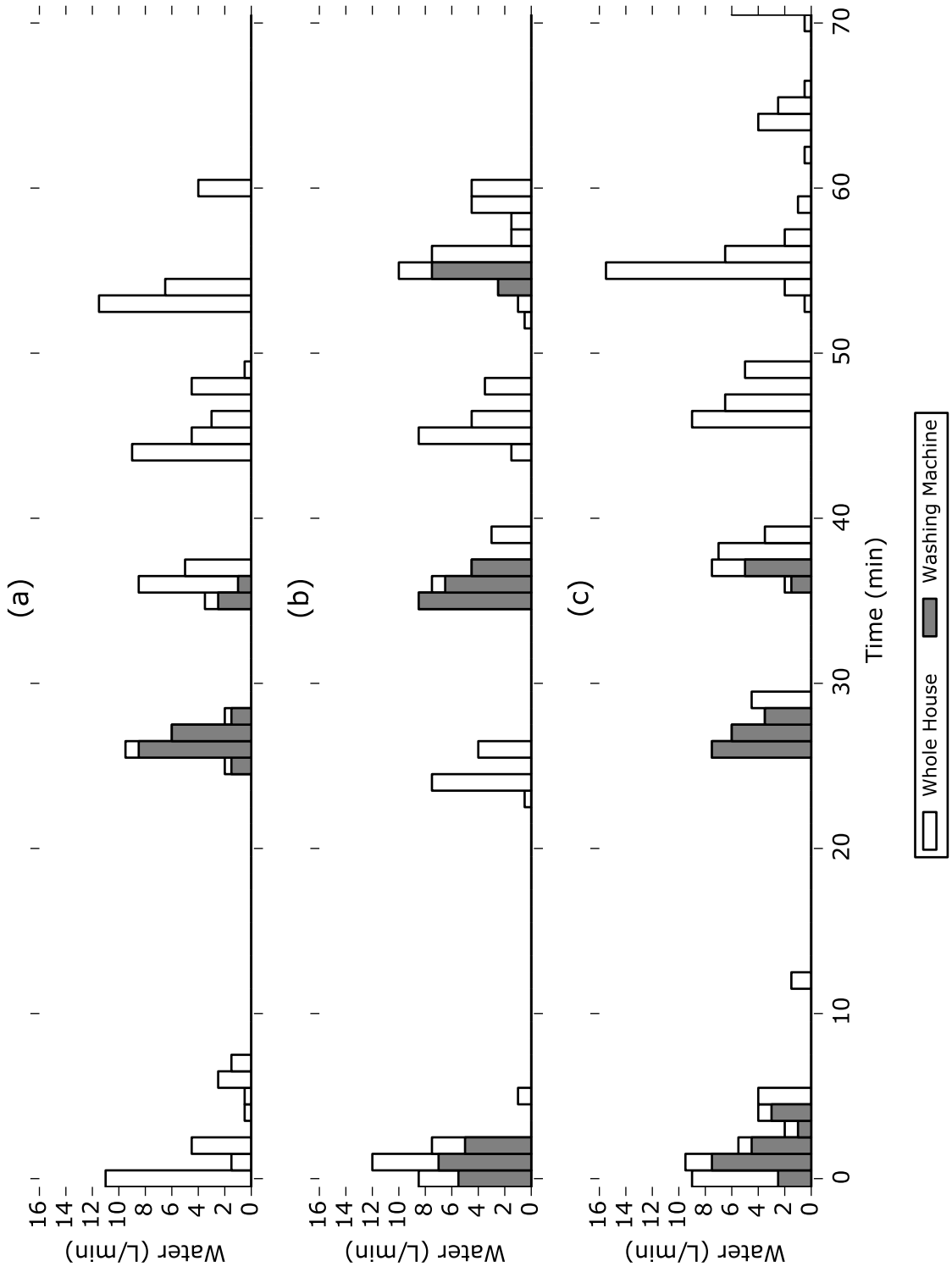

Figure 4.8: Output of second-order model on three washing machine examples. Only two (a) or three (b, c) of the five expected water-use actions are captured.

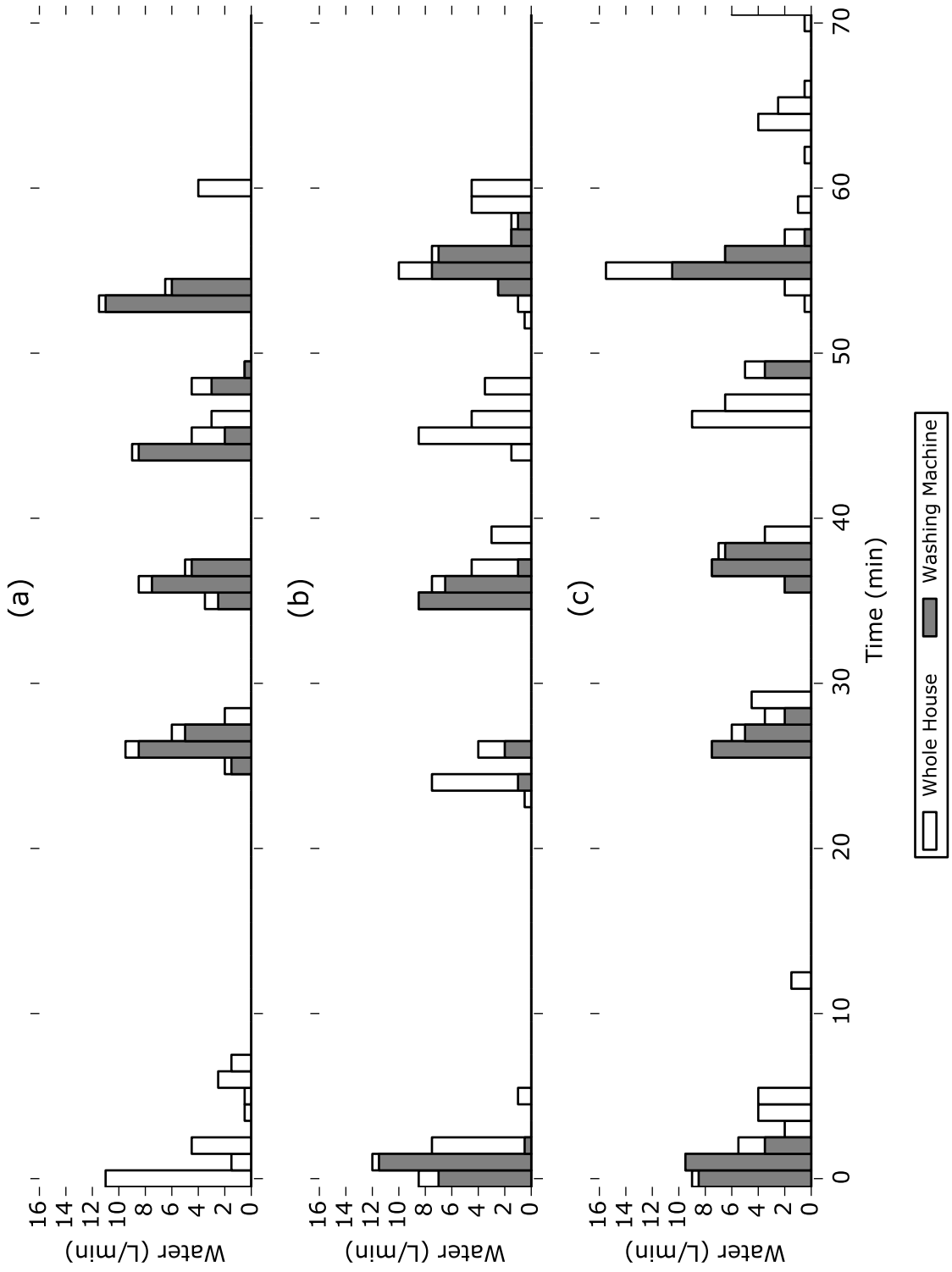

Figure 4.9: Output of third-order model on three washing machine examples. Four (a, b) or five (c) of the five expected water-use actions are at least partially captured.

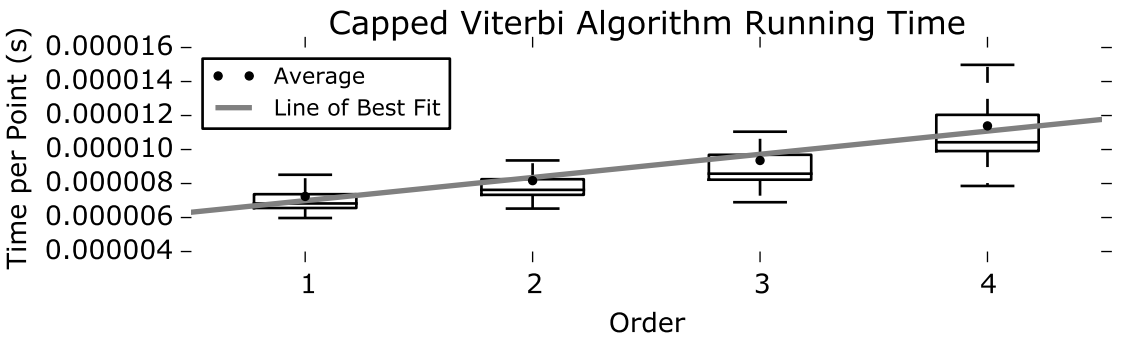

Figure 4.10: Box plots showing the running times of the Capped Viterbi Algorithm for the experiments in Figure 4.11 and Figure 4.12. Times are normalized by the number of points in the data series being labelled.

to the dishwasher, this is actually beneficial for the washing machine. Since the water-use actions take longer and consume more water, there is more room for variation. Taking coarser readings helps smooth these differences out. Figure 4.10 shows comparable running times to Figure 4.4.

Here we extracted 66 activities split into 10 folds with 6 or 7 activities each. Figure 4.11 shows the entire extraction process outlined in §3.2 applied to one of the gallon granularity activities. The dishwasher current PMF for this portion of the dataset is shown in Figure 4.11a (c.f. Figure 3.6). These states are applied to our example in Figure 4.11b. States are labelled on the corresponding whole house water readings in Figure 4.11c.

The results of the Capped Viterbi Algorithm on this example are shown in Figure 4.12. Results of the second-, third- and fourth-order models are shown in Figure 4.12a, Figure 4.12b, and Figure 4.12c, respectively. As above, we do not have ground truth for this example, but it is easier to see a clearer pattern here. Even though there is more noise than the examples in Figure 4.8 and Figure 4.9, the models are able to learn that (at this granularity) a water-use action for the washing machine consists of two consecutive one-gallon pulses. Consistent with all other experiments, the third-order model performs best.

4.3 Summary

In this chapter we conducted quantitative and qualitative analyses of the Capped Viterbi Algorithm on the dishwasher and washing machine in AMPds. We found in all cases that the third-order model performed best. This model is able to strike a balance between having too little previous information about actions (second-order) and over-fitting activities (fourth-order). It succeeds in the goal of learning how water-use actions and electricity-use actions correlate using the extracted activities. For machines that may have longer actions, the Capped Viterbi Algorithm has the potential to scale up to higher orders.

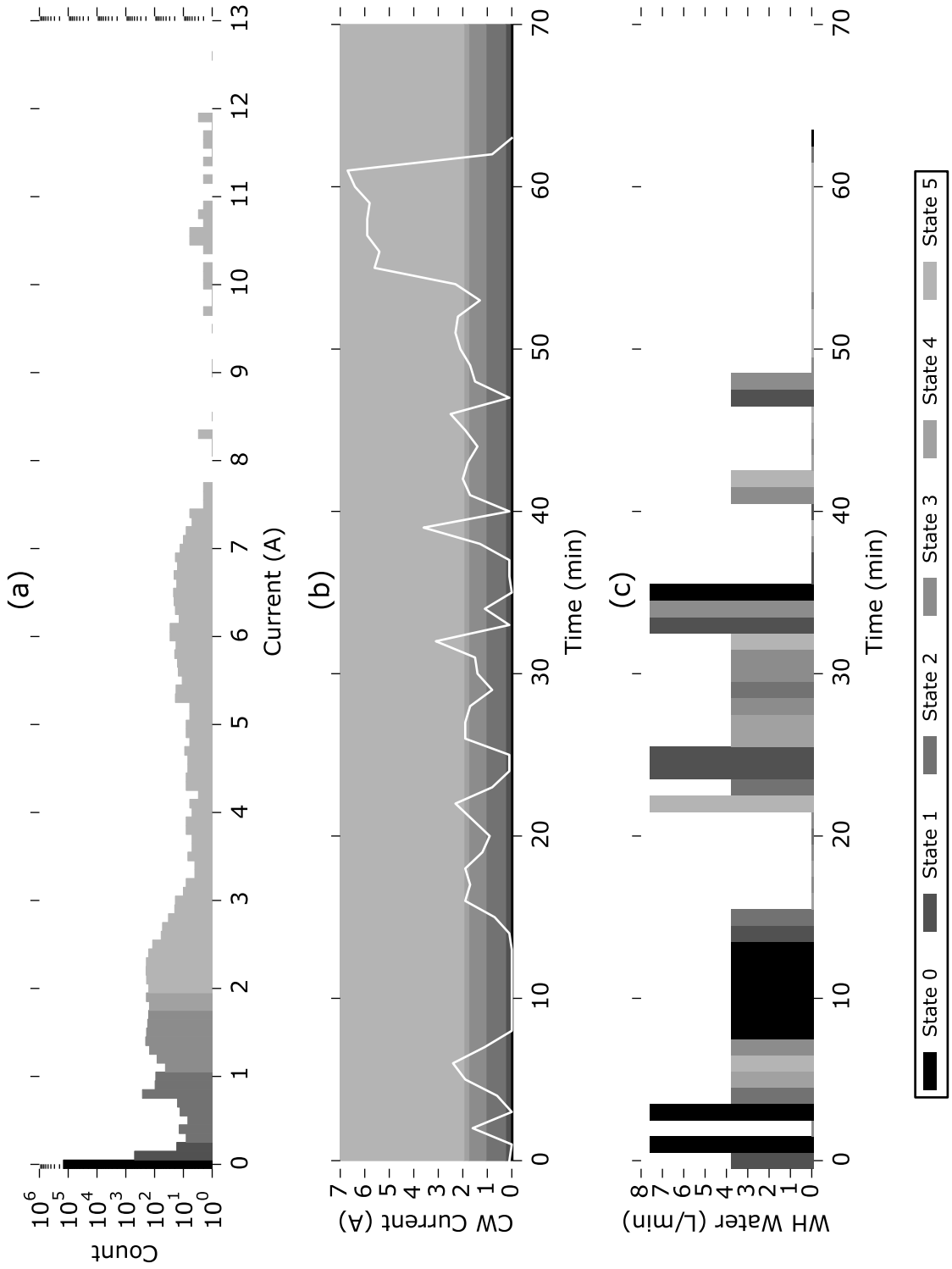

Figure 4.11: The extraction process for a gallon granularity example including the washing machine current PMF (a), the states applied to the current reading (b), and the corresponding whole house water readings (c).

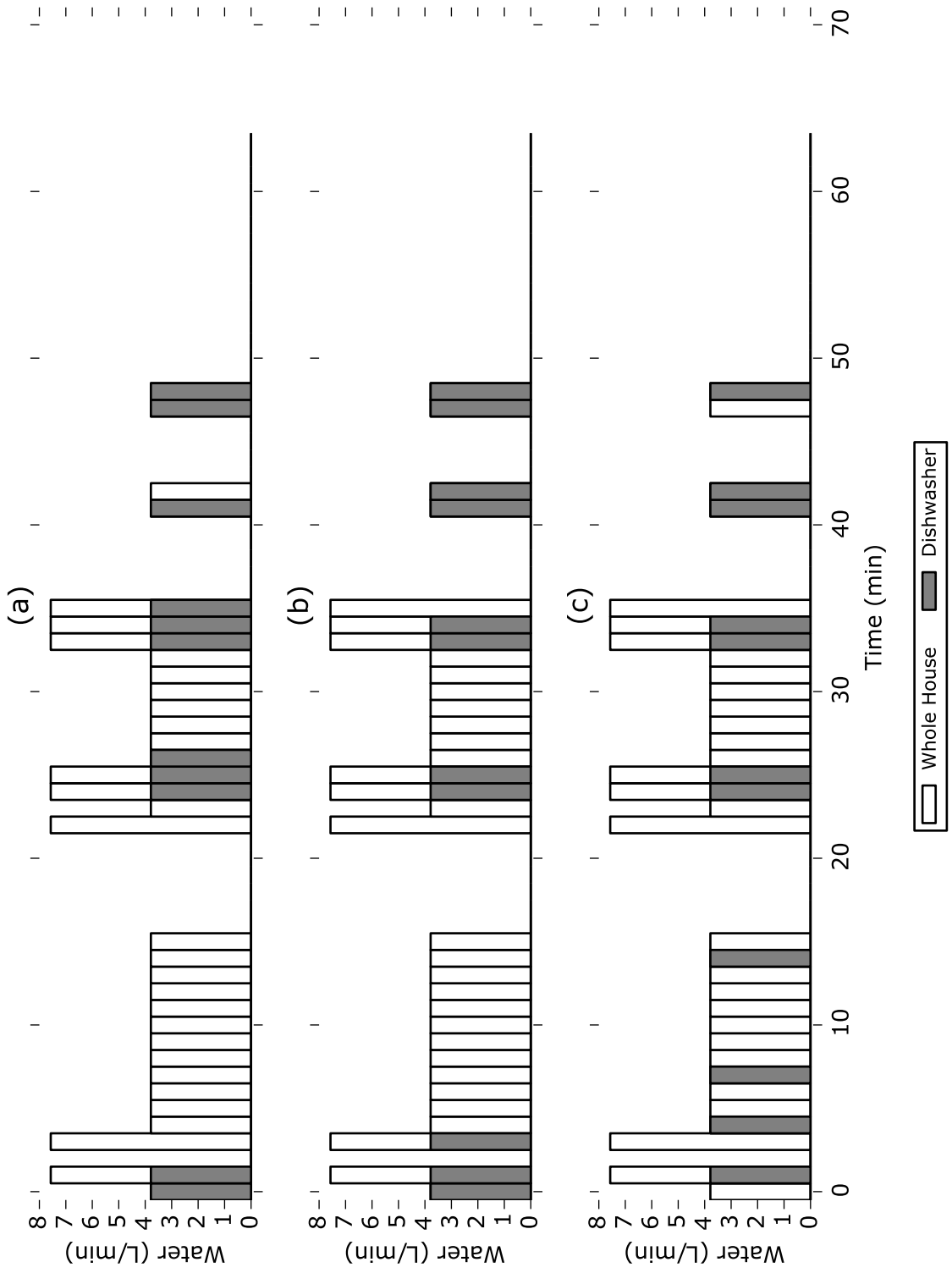

Figure 4.12: Results of running the Capped Viterbi Algorithm on the example from Figure 4.11 including second-order (a), third-order (b), and fourth-order (c).

Chapter 5

Conclusions

The field of water disaggregation has tended towards studies that focus on identifying low-level differences between similar fixtures. In doing so, non-standard sensors must be introduced to collect additional data. By situating the problem in terms of a hierarchy (§1.1), we were able to pinpoint the level to which disaggregated water information is helpful to homeowners. Namely, to inform sustainable practices, one only needs to know the amount of consumption attributed to each type of fixture, not to each specific fixture. For some cases, knowing only the machine-use versus human-use may be enough.

In this thesis we presented a model for disaggregating household machine water consumption using information from electricity disaggregation in addition to the water data. Using data from pre-existing sensors prevents the need for extraneous information. It also allows a system to be built that does not require tuning by an expert.

Integrating our more accurate method for handling machine water-use with previous work that has focused on human water-use could build a system for providing a comprehensive breakdown of household water consumption. In particular, looking towards other environmental factors for fixtures that do not also consume electricity [39, 40] could help complete the full picture of the hierarchy in Figure 1.1. For techniques that iteratively decompose the water reading by labelling the residual at each step [12], our algorithm could be used as the first pass to simplify disaggregating the rest of a house. We also alluded to the possibility of leveraging timer information or outdoor water-use restrictions to disaggregate lawn sprinklers in §1.1.

We have shown the need for standardized datasets like AMPds [33]. This allows the researcher to focus on the task of model design and evaluation, while not being bogged down by issues of inconsistency. It would be nice to have data from more households to test our models against. Any datasets incorporating water and electricity data must be collected using properly calibrated sensors. The data must be cleaned and presented in a format that allows different groups to test different models in the same fashion. Additionally, having submetered ground truth for water data to test against would be a plus.

Our disaggregator, the new Capped Viterbi Algorithm outlined in §3.3, currently relies on the extraction of whole activities, as described in §3.2. Activities can be labelled very quickly, but this task cannot be done until all the required data points have been gathered. This allows a homeowner to see a machine's water consumption immediately after it is finished running, but not during a cycle. In the future, we would like to evaluate the potential for a real-time system using a greedy technique, akin to the one used in [32].

Chapter 4 demonstrates the importance of collecting data at the proper granularity. A finer granularity allows more subtleties to be picked up, but also results in greater variability in readings. For both the dishwasher and the washing machine, tests were conducted at both the level of half-litre pulses and gallon pulses. In the case of the dishwasher, performance was better with half-litre pulses, due to the water-use actions being under a gallon. Conversely, the disaggregator performed better on the washing machine data with gallon pulses, due to the water-use actions being much greater than half a litre. It would be interesting to see tests carried out on data purposefully collected at a wider range of granularities.

Besides the granularity of the pulses, there is also the frequency of the pulses. We feel that reporting at any lower frequency would not provide a detailed enough breakdown of water-use for a homeowner. This could make our third-order model insufficient for machines that have more drawn out actions. For example, in a home with multiple dishwashers, the electricity might only be jointly disaggregated due to the similarity in the patterns. In rare instances where both machines are running at once, there could be more complex compound actions. With data like this, one could try even higher-order models. The Capped Viterbi Algorithm shows promise for scaling up efficiently in terms of the order of the model.

For other future work, we would like to see the applicability of our model to other types of parallel data. As an example, AMPds includes data for natural gas. We would like to see if we could use the same principles to leverage electricity data to disaggregate natural gas readings. In the case of the instant hot water unit, there is also submetered water data. This opens up the door to the potential for leveraging water data to disaggregate natural gas readings.

Down the road we foresee this research being applicable to building a larger NILM system. By restricting disaggregation to broader categories, privacy issues are reduced when sharing data. This means this information could be used not only by individual homeowners, but also by utility companies to offer monetary incentives for sustainable practices. Averaging values across neighbourhoods or suburbs could also allow municipalities to motivate conservation by showing residents how they compare to each other.

Bibliography

[1] Nipun Batra, Manoj Gulati, Amarjeet Singh, and Mani B Srivastava. It's different: Insights into home energy consumption in India. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, BuildSys'13, pages 31–38, New York, NY, USA, 2013. ACM.

[2] Stephen P Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[3] Tim Campbell, Eric Larson, Gabe Cohn, Jon Froehlich, Ramses Alcaide, and Shwetak N Patel. WATTR: A method for self-powered wireless sensing of water activity in the home. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, UbiComp '10, pages 169–172, New York, NY, USA, 2010. ACM.

[4] J Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.

[5] Heather Chappells, Will Medd, and Elizabeth Shove. Disruption and change: drought and the inconspicuous dynamics of garden lives. *Social & Cultural Geography*, 12(7):701–715, 2011.

[6] Feng Chen, Jing Dai, Bingsheng Wang, Sambit Sahu, Milind Naphade, and Chang-Tien Lu. Activity analysis based on low sample rate smart meters. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 240–248, 2011.

[7] Allen Chun. Flushing in the future: The supermodern Japanese toilet in a changing domestic culture. *Postcolonial Studies*, 5(2):153–170, 2002.

[8] W A Church and K W Gale. What's wrong with adding one. *Statistical Research Reports*, 1994.

[9] Rob de Loë, Reid Kreutzwiser, and Liana Moraru. Adaptation options for the near term: climate change and the canadian water sector. *Glob. Environ. Change*, 11(3):231–245, October 2001.

[10] DENT Instruments, Inc. *User's Manual: PowerScout 3, PowerScout 18, ViewPoint*, 2010.

[11] W B DeOreo, J P Heaney, and P W Mayer. Flow trace analysis to assess water use. *Journal of the American Water Works Association*, 88(1):79–90, 1996.

[12] Haili Dong, Bingsheng Wang, and Chang-Tien Lu. Deep sparse coding based recursive disaggregation model for water conservation. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, pages 2804–2810, 2013.

[13] Elster Metering Limited. *V100 Volumetric Cold Water Meters: Product Specification*, 2008.

[14] Environment Canada. 2011 municipal water use report: Municipal water use 2009 statistics. Technical report, 2011.

[15] R Warren Flint. The sustainable development of water resources. *Journal of Contemporary Water Research and Education*, 127(1):6, 2010.

[16] James Fogarty, Carolyn Au, and Scott E Hudson. Sensing from the basement: A feasibility study of unobtrusive and low-cost home activity recognition. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, UIST '06, pages 91–100, New York, NY, USA, 2006. ACM.

[17] Jr Forney, G.D. The viterbi algorithm. *Proc. IEEE*, 61(3):268–278, 1973.

[18] Frigidaire. *Washer Operating Instructions: PN 134374500A (0411)*.

[19] Jon Froehlich, Leah Findlater, and James Landay. The design of eco-feedback technology. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 1999–2008, New York, NY, USA, 2010. ACM.

[20] Jon Froehlich, Leah Findlater, Marilyn Ostergren, Solai Ramanathan, Josh Peterson, Inness Wragg, Eric Larson, Fabia Fu, Mazhengmin Bai, Shwetak Patel, and James A Landay. The design and evaluation of prototype eco-feedback displays for fixture-level water usage data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2367–2376, New York, NY, USA, 2012. ACM.

[21] Jon Froehlich, Eric Larson, Tim Campbell, Conor Haggerty, James Fogarty, and Shwetak N Patel. HydroSense: Infrastructure-Mediated Single-Point sensing of Whole-Home water activity. In *Proceedings of the 11th International Conference on Ubiquitous Computing*, pages 235–244, 2009.

[22] Zoubin Ghahramani. An introduction to hidden markov models and bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(01):9–42, 2001.

[23] Peter H Gleick. *Water futures: A review of global water resources projections*. Pacific Institute for Studies in Development, Environment and Security, 1999.

[24] I J Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3-4):237–264, 1953.

[25] G W Hart. Prototype nonintrusive appliance load monitor. Technical report, MIT Energy Laboratory and Electric Power Research Institute Technical Report, 1985.

[26] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Linear methods for classification. In *The Elements of Statistical Learning*, Springer Series in Statistics, pages 101–137. Springer New York, 2009.

[27] L J Heyer, S Kruglyak, and S Yooseph. Exploring expression data: identification and analysis of coexpressed genes. *Genome Res.*, 9(11):1106–1115, 1999.

[28] Hyungsul Kim, Manish Marwah, Martin F Arlitt, Geoff Lyon, and Jiawei Han. Unsupervised disaggregation of low frequency power measurements. In *SDM*, volume 11, pages 747–758, 2011.

[29] Younghun Kim, Thomas Schmid, Zainul M Charbiwala, Jonathan Friedman, and Mani B Srivastava. NAWMS: Nonintrusive autonomous water monitoring system. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, SenSys '08, pages 309–322, New York, NY, USA, 2008. ACM.

[30] E C Larson. Consumer centered calibration in end-use water monitoring. In *NILM Workshop 2014*, 2014.

[31] Eric Larson, Jon Froehlich, Tim Campbell, Conor Haggerty, Les Atlas, James Fogarty, and Shwetak N Patel. Disaggregated water sensing from a single, pressure-based sensor: An extended analysis of HydroSense using staged experiments. *Pervasive and Mobile Computing*, 8(1):82–102, 2012.

[32] Stephen Makonin, Ivan V Bajic, and Fred Popowich. Efficient sparse matrix processing for nonintrusive load monitoring (NILM). In *2nd International Workshop on Non-Intrusive Load Monitoring*, 2014.

[33] Stephen Makonin, Fred Popowich, Lyn Bartram, Bob Gill, and Ivan V Bajic. AMPds: A public dataset for load disaggregation and Eco-Feedback research. In *Electrical Power & Energy Conference*, pages 1–6, 2013.

[34] P W Mayer, W B DeOreo, E M Opitz, J C Kiefer, W Y Davis, and others. *Residential End Uses of Water*. American Water Works Association, 1999.

[35] Oracle Corporation. Smart metering for water utilities. Technical report, 2009.

[36] L Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[37] C Schantz, J Donnal, S Leeb, P N Marimuthu, and S Habib. WaterWOLF: Water watch on load flow. In *Urban Water II*, volume 1 of *WIT Transactions on The Built Environment*, pages 125–136, Southampton, UK, 2014. WIT Press.

[38] C Schantz, J Donnal, B Sennett, M Gillman, S Muller, and S Leeb. Water Non-Intrusive load monitoring. *IEEE Sensors Journal*, PP(99):1, 2014.

[39] Vijay Srinivasan, John Stankovic, and Kamin Whitehouse. WaterSense: Water flow disaggregation using motion sensors. In *Proceedings of the Third ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, BuildSys '11, pages 19–24, New York, NY, USA, 2011. ACM.

[40] Vijay Srinivasan, John Stankovic, and Kamin Whitehouse. FixtureFinder: Discovering the existence of electrical and water fixtures. In *Proceedings of the 12th International Conference on Information Processing in Sensor Networks*, IPSN '13, pages 115–128, New York, NY, USA, 2013. ACM.

[41] Statistics Canada Environmental Accounts. Human activity and the environment: Freshwater supply and demand in Canada. Technical report, 2010.

[42] Tim L M van Kasteren, Gwenn Englebienne, and Ben J A Kröse. Hierarchical activity recognition using automatically clustered actions. In *Ambient Intelligence*, Lecture Notes in Computer Science, pages 82–91. Springer Berlin Heidelberg, 2011.

[43] A J Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inf. Theory*, 13(2):260–269, 1967.

[44] Bingsheng Wang, Feng Chen, Haili Dong, Arnold P Boedihardjo, and Chang-Tien Lu. Signal disaggregation via sparse coding with featured discriminative dictionary. In *IEEE 12th International Conference on Data Mining*, pages 1134–1139. IEEE, 2012.

[45] World Water Assessment Programme (United Nations). *Water for People, Water for Life: A Joint Report by the Twenty-three UN Agencies Concerned with Freshwater*. UNESCO Pub., 2003.