An Unsupervised Training Method for Non-intrusive Appliance Load Monitoring[☆]

Oliver Parson, Siddhartha Ghosh, Mark Weal, Alex Rogers

Electronics and Computer Science
University of Southampton, Hampshire, SO17 1BJ, UK

Abstract

Non-intrusive appliance load monitoring is the process of disaggregating a household's total electricity consumption into its contributing appliances. In this paper we propose an unsupervised training method for non-intrusive monitoring which, unlike existing supervised approaches, does not require training data to be collected by sub-metering individual appliances, nor does it require appliances to be manually labelled for the households in which disaggregation is performed. Instead, we propose an approach which combines a one-off supervised learning process over existing labelled appliance data sets, with an unsupervised learning method over unlabelled household aggregate data. First, we propose an approach which uses the Tracebase data set to build probabilistic appliance models which generalise to previously unseen households, which we empirically evaluate through cross validation. Second, we use the Reference Energy Disaggregation Data set to evaluate the accuracy with which these general models can be tuned to the appliances within a specific household using only aggregate data. Our empirical evaluation demonstrates that general appliance models can be constructed using data from only a small number of appliances (typically 3-6 appliances), and furthermore that 28-99% of the remaining behaviour which is specific to a single household can be learned using only aggregate data from existing smart meters.

[☆]A preliminary version of this work appeared in [31]

Email addresses: osp@ecs.soton.ac.uk (Oliver Parson), sg2@ecs.soton.ac.uk (Siddhartha Ghosh), mjw@ecs.soton.ac.uk (Mark Weal), acr@ecs.soton.ac.uk (Alex Rogers)

Keywords: Machine learning, Bayesian networks, Unsupervised learning, Computational sustainability, Smart grid

1. Introduction

With many countries aiming to considerably reduce their annual carbon emissions by 2050, energy conservation has become an issue of national importance [9]. To this end, ageing electricity infrastructure is undergoing a transition towards the smart grid, in which a high degree of monitoring will enable the flow of information between the points of energy generation and consumption [36]. As part of this transition, smart meters are currently being deployed on national scales [10], and will soon be collecting vast amounts of domestic electricity consumption data. However, smart meters will only make information visible regarding a household's total electricity consumption, while the energy consumption of individual appliances will remain invisible to the household's occupants. Without such personalised feedback, each household's occupants will be left to guess which appliances and activities consume the most energy, which a recent review of the literature has shown to often be a poor estimation of the true energy breakdown [14]. This raises a key artificial intelligence challenge, regarding how personalised useful insight can be produced entirely automatically from millions of households' smart meter data [34].

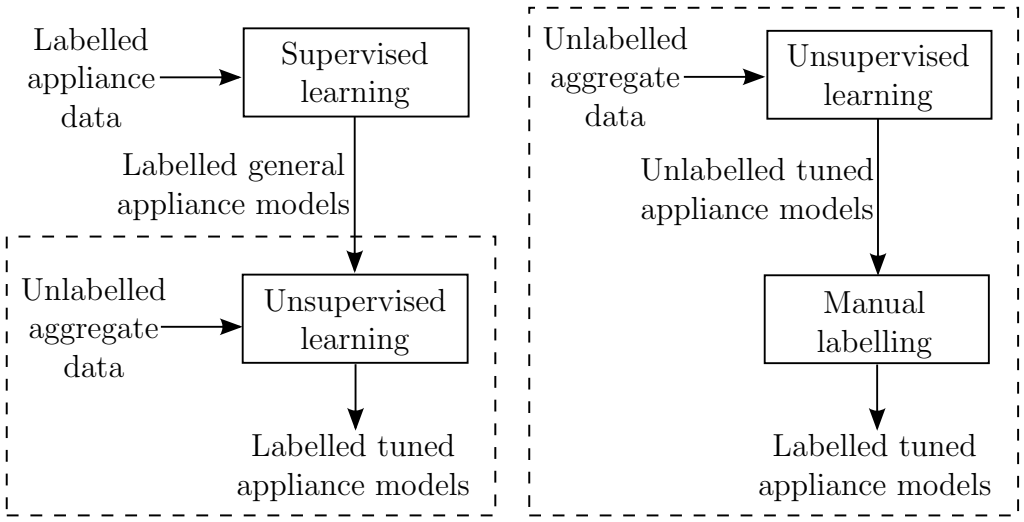
To this end, non-intrusive appliance load monitoring (NIALM), or energy disaggregation, aims to break down a household's aggregate electricity consumption as collected by a smart meter into individual appliances [18]. Studies have shown that providing a household's occupants with a personalised breakdown of appliance energy consumption allows them to take steps towards reducing their total energy consumption [8]. A recent review of appliance-specific feedback literature has indicated that such information can reduce household energy consumption by 14% on average [12]. Furthermore, even greater reductions can be achieved if the disaggregated appliance data is used to produce actionable feedback [1]. Although in general smart meters transmit only 15-30 minute aggregate data to the utility for billing purposes, many smart meters also transmit 10 second power data over the home area network (e.g. UK smart meters [11]). Such information can be consumed by authenticated devices and either processed locally or uploaded to cloud storage. However, many existing approaches require a sampling rate

in the order of kHz [17, 4, 5], and therefore are not applicable to such smart meter data. In summary, NIALM can be formulated as a machine learning problem, in which 10 second smart meter electricity data is required to be separated into the contributing appliances.

Recent contributions to this problem fall into three categories. The first use supervised methods which assume that sub-metered appliance training data are available from the household in which disaggregation is to be performed. One approach to collect this data is to install individual appliance sub-meters [5, 13, 26]. However, this assumption dramatically decreases the scalability of such systems due to the inherent costs and time consuming nature of installing individual appliance meters, and furthermore this process renders NIALM unnecessary since appliance level data is already collected. Hart [18] proposed an alternative approach in which the appliances are operated sequentially allowing individual appliance signatures to be extracted from aggregate measurements. Weiss et al. [38] extended this approach by using smart phones to label each individual appliance's operation. However, it is impractical to require the occupants of every household in a country to carry out such system training in every home.

The second category of existing approaches uses unsupervised disaggregation methods in which no prior knowledge of the appliances is assumed. Such approaches have demonstrated how appliance parameters can be learned after collecting aggregate data for a suitable period, ranging from 1.5 hours [16], to 5-26 days [40, 22, 2] or even 6 months [20]. However, since these methods only learn parameters for a set of classes (corresponding to different appliances), they are unable to assign labels (appliance names) to each class. As a result, only the distinction between appliances is unsupervised, and these approaches still require a manual labelling process in which each learned class is matched to an appliance label by a domain expert. Most recently, attempts have been made to avoid the manual labelling process by automatically encoding general appliance information in a Bayesian inference framework [19]. However, this method has only been demonstrated for households containing up to 5 appliance types, and the large state space introduced by 15-20 appliance types is likely to cause the inference process to be unable to distinguish between appliances. Therefore, none of these methods scale automatically to realistic previously unseen homes.

A third category has been suggested which would require the collection of an exhaustive signature database of multiple signatures for possible appliances [25, 24]. However, the sheer number of different instances of each



(a) Our proposed approach (b) Existing approaches [20, 22]

Figure 1: Comparison of our proposed approach and existing approaches. The dashed box represents processes which must be repeated for each household in which disaggregation is performed.

appliance type makes this approach prohibitively expensive and time consuming. Furthermore, it is unlikely a classifier would be able to distinguish between different appliance instances in such a crowded feature space. For these reasons, the construction of such an appliance database has never been attempted [35].

In addition to these three categories of approaches which are applicable to 10 second resolution data, it is also worth considering approaches that have been applied to other resolutions of data. Kolter et al. [21] applied an approach based on sparse coding to data at 1 hour resolution. The authors showed that their approach was able to learn appliance models which generalise between households. However, this approach bases the disaggregation largely on the energy consumed over a duration of time, and the hour of day and day of week in which the energy was consumed. As a result, it does not model the strong dependency between sequential measurements in 10 second power data. Furthermore, sparse coding would require a large number of dimensions to represent such data, and as a result is prone to overfitting when training data is limited. Therefore, sparse coding is not an ideal approach to be applied to data collected by smart meters.

Clearly, none of the discussed approaches present realistic methods which can operate entirely automatically on 10 second resolution data from a previously unseen home, and therefore none of these approaches are applicable to disaggregate smart meter data at national scales. To address this shortcoming, we propose a method for training NIALM systems which uses information about appliances and generalises to previously unseen households. Crucially, this means our approach does not require either sub-metered training data or a manual labelling phase for each household in which disaggregation is performed. Figure 1 shows the distinction between our approach and existing work, in which the processes which must be repeated for each new household are highlighted by the dashed box. Such generalisable appliance information is available in appliance monitoring data sets, such as the Tracebase data set [33]. By modelling appliances as hidden Markov models (HMMs), we learn a general appliance behavioural model which will generalise to previously unseen households. We then use an unsupervised method by which these general appliance models can be tuned to the specific appliance instances in a previously unseen household using only aggregate data. We evaluate our approach using the Reference Energy Disaggregation Data set (REDD) [23]. Finally, we use our approach to determine energy efficiency feedback which could be provided to a household's occupants, although the tuned appliance models could also be used to allow existing disaggregation methods to be applied to new households without a manual training phase. Our contributions can be summarised as follows:

- We propose a hierarchical approach which models multiple appliances of the same type based on a Bayesian treatment of HMMs [15]. As such, the parameters of multiple HMMs can be combined to form a general model of an appliance. We show that only 3-6 examples of an appliance type are required to sufficiently generalise to a previously unseen appliance.

- We provide a method by which the general appliance models can be tuned to the appliances within a specific household using only aggregate data. We show that models tuned using only aggregate data outperform the general models, and in some cases perform comparably to models learned using sub-metered data from the test appliance. Furthermore, we show that our tuning method outperforms the state of the art which uses factorial HMMs to tune appliance models.

- We give a number of examples of the personalised feedback which tuned appliances models can be used to provide. We show that by using only general appliance models and smart meter data, advice can be given to household occupants regarding the operating energy efficiency of the appliances and whether it is cost effective to replace them.

The remainder of this paper is structured as follows. In Section 2 we describe how appliance models can be learned from reference data sets that will generalise to previously unseen households. In Section 3 we describe how these general models can be tuned to specific appliance instances using only aggregate data from the test house. In Section 4 we describe the additional benefit tuned appliance models provide beyond their primary application as the input to a disaggregation system. Finally, we conclude in Section 5.

2. Building Generalisable Appliance Models

We now describe our method for learning general appliance models which will generalise to previously unseen appliances of the same type. The aim is to learn distributions over the model parameters for each appliance, such that both the mean and variance around each appliance parameter is derived from data. This process is effective as it allows tight distributions to be learned over appliance parameters which are similar for different appliance instances, and broad distributions to be learned when parameters vary greatly between different instances. In general, the most important factor is to ensure that the learned states align between different appliance instances, which we demonstrate through the Bayesian framework described in Section 2.1.

Throughout this section, we use a running example of the refrigerator to provide some intuition into the model choices and role of various parameters. The remainder of this section is structured as follows. First, we propose a hierarchical approach to model multiple appliance instances of the same type. Next, we show how common signatures which capture the general behaviour of an appliance can be extracted from different appliances' power data. Finally, we give an empirical evaluation which demonstrates the benefit of generalising over multiple appliances of the same type.

We adopt a hierarchical approach to model multiple appliances of the same type, as shown by Figure 2. In this model, we represent an appliance type (e.g. refrigerator) as a distribution from which appliance instances (e.g. Bosch Logixx KSV36AW41G refrigerator) are drawn. As such, the appliance type represents any behaviour which is common to all instances of that

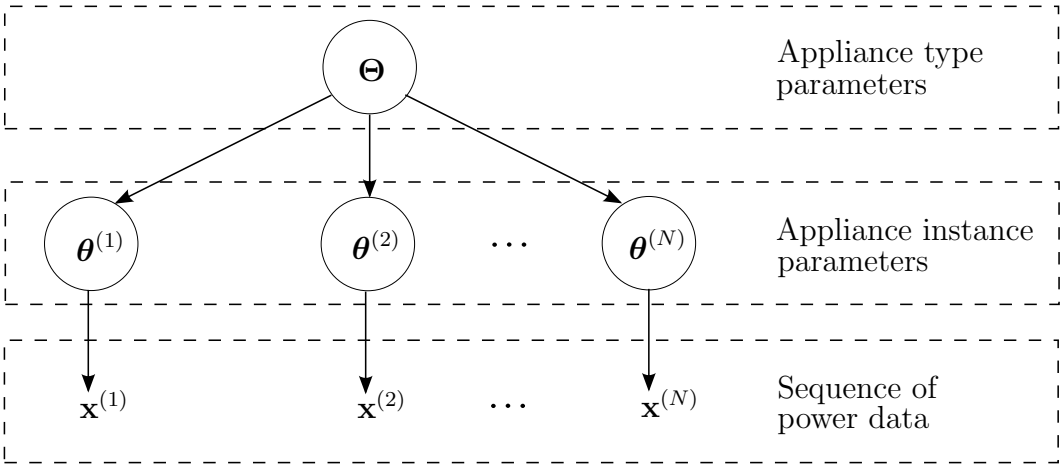


Figure 2: Hierarchical model of an appliance type $\boldsymbol{\Theta}$

type, while an appliance instance also represents behaviour which is specific to that single instance and its usage. Furthermore, we observe sequences of power data from each appliance instance. Therefore, the aim is to infer the parameters of an appliance type, $\boldsymbol{\Theta}$, from sequences of power data, $\mathbf{x}^{(n)} = \{x_1, \ldots, x_T\}$, generated by individual appliance instances described by parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(N)}\}$, where n is one of N appliance instance indices.

In order to learn the appliance type parameters, we first estimate the parameters of each appliance instance from a sequence of power readings as described in the following section. We then describe a method for generalising over these parameters in Section 2.2.

2.1. Appliance Instance Parameter Estimation using Hidden Markov Models

We adopt a hidden Markov model (HMM) representation for household appliances [15]. A HMM consists of a Markov chain of discrete, latent variables (representing the operational state of an appliance) and a sequence of continuous, observed variables (representing the power demand of an appliance), each of which is dependent upon one of the discrete variable's state. Figure 3 gives the structure of a HMM as a Bayesian network, where the discrete, latent variables are represented by the sequence $z_1, \ldots, z_T$, and the continuous, observed variables are represented by the sequence $x_1, \ldots x_T$, over a time sequence of length T. The value of each discrete variable z_t corresponds to one

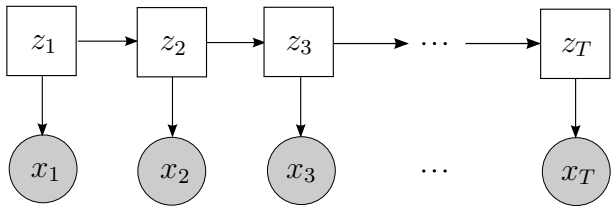
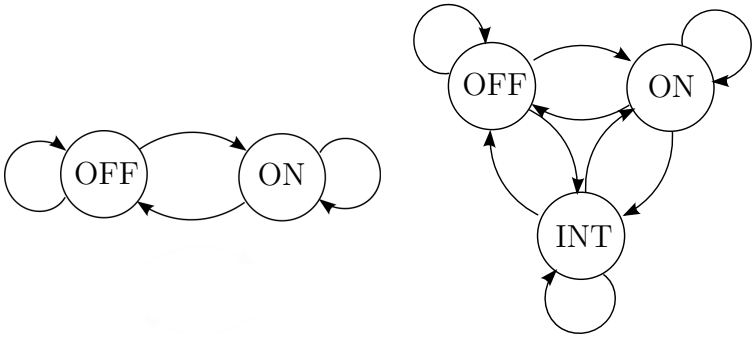


Figure 3: A hidden Markov model. Unshaded squares represent discrete, latent variables and shaded circles represent continuous, observed variables.



(a) 2 state model (b) 3 state model

Figure 4: Appliance state models

of K states (e.g. on, off) as shown in Figure 4, while each continuous variable can be either zero or any positive real number (e.g. 100.5 W), since appliances only consume energy. For the sake of clarity, we omit the appliance instance index $^{(n)}$ throughout Section 2.1.

A hidden Markov model can be completely defined by the following three parameters. First, the probability of each state, k, of the hidden variable at $t = 1$ can be represented by the vector $\boldsymbol{\pi}$ such that:

$$\pi_k = p(z_1 = k) \tag{1}$$

In the case of the refrigerator, this corresponds to the probability that the appliance is either on or off at the start of the data sequence.

Second, the probability of a transition from state i at $t - 1$ to state j at t can be represented by the matrix $\mathbf{A}$ such that:

$$A_{i,j} = p(z_t = j | z_{t-1} = i) \tag{2}$$

In the case of the refrigerator, this corresponds to the probability that the

appliance has either turned on, turned off, or remained in the same state between consecutive power measurements.

Third, the emission probabilities for $\mathbf{x}$ are described by a function governed by parameters $\boldsymbol{\phi}$, which in our case is assumed to be Gaussian distributed such that:

$$x_t|z_t, \boldsymbol{\phi} \sim \mathcal{N}(\mu_{z_t}, \tau_{z_t}) \tag{3}$$

where $\boldsymbol{\phi}_k = \{\mu_k, \tau_k\}$, and μ_k and τ_k are the mean and precision of state k's Gaussian distribution. We use a Gaussian distribution since it has previously been shown to provide a good fit of appliance power demand [20]. Although an appliance's power demand is strictly positive, we found that the Gaussian distribution's support for negative power demands is negligible for most appliances. However, it is worth noting that other distributions could also be used if a strictly positive (e.g. gamma distribution) or a multi-modal (e.g. a mixture of Gaussians) distribution were required. In the case of the refrigerator, the off state emission distribution will likely be a very high precision distribution centered around 0 W, while the on state distribution will be a slightly lower precision distribution centered around approximately 100 W. Both distributions are expected to be of relatively high precision since this the precision parameter represents only small fluctuations in the appliance's power demand.

Equations 1, 2 and 3 can be used to calculate the joint likelihood of a hidden Markov model:

$$p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) = p(z_1|\boldsymbol{\pi}) \prod_{t=2}^{T} p(z_t|z_{t-1}, \mathbf{A}) \prod_{t=1}^{T} p(x_t|z_t, \boldsymbol{\phi}) \tag{4}$$

where the set of model parameters is represented by $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\phi}\}$. These parameters are not known a priori and are learned from data.

We adopt a Bayesian approach to learn the parameters of HMMs in which prior distributions are placed over the model parameters. A Bayesian approach is required in this scenario, since it ensures that the states learned for one instance of an appliance type correspond to the same states learned from a different instance of the same appliance type. For example, it ensures that the *spin* state of washing machine A corresponds to the *spin* state of washing machine B, etc. By placing conjugate priors over the model parameters, we ensure that both the priors and posteriors belong to the same family of distributions. We now describe the prior and posterior distributions over the model parameters, which for the sake of clarity, we use a hat to denote the

hyperparameters of the prior distributions (e.g. $\hat{\boldsymbol{\alpha}}$), and a tilde to denote the parameters of the posterior distributions (e.g. $\tilde{\boldsymbol{\alpha}}$).

The initial probabilities follow a categorical distribution, for which the conjugate prior is the Dirichlet distribution:

$$\boldsymbol{\pi} \sim \text{Dir}(K, \hat{\boldsymbol{\alpha}}) \tag{5}$$

where Dir is the Dirichlet distribution parameterised by the number of categories, K, and the concentrations parameters, $\hat{\boldsymbol{\alpha}}$. We denote the parameters of the posterior distribution as $\tilde{\boldsymbol{\alpha}}$. In the case of the refrigerator, we have little a priori information regarding the initial distribution, and so a uniform prior distribution is used.

Similarly, each row, i, of the transition matrix also follows a categorical distribution:

$$\mathbf{A}_i \sim \text{Dir}(K, \hat{\mathbf{C}}_i) \tag{6}$$

where Dir is the Dirichlet distribution parameterised by the number of categories, K, and a vector of concentrations parameters, $\hat{\mathbf{C}}_i$. We denote the posterior parameters as $\tilde{\mathbf{C}}_i$. In the case of the refrigerator, there is a sufficient amount of training data available in existing appliance data sets, and so a uniform prior is also a sufficient distribution.

Finally, the emission variables are Gaussian distributed, for which a conjugate prior is the Gaussian-gamma distribution [30]:

$$\mu_k \sim \mathcal{N}(\hat{\lambda}_k, \hat{r}_k) \tag{7}$$

$$\tau_k \sim \text{Gamma}(\hat{\beta}_k, \hat{w}_k) \tag{8}$$

where $\mathcal{N}$ is the Gaussian distribution parameterised by mean, $\hat{\lambda}_k$, and precision, $\hat{r}_k$, and Gamma is the gamma distribution parameterised by shape, $\hat{\beta}_k$, and scale, $\hat{w}_k$. We denote the respective parameters of each posterior distribution as $\tilde{\lambda}_k$ and $\tilde{r}_k$, and $\tilde{\beta}_k$ and $\tilde{w}_k$.

It is crucial to incorporate domain knowledge via these hyperparameters to ensure the posterior states correspond between different appliance instances. In the case of the refrigerator, $\hat{\lambda}_{off}$ would be 0 W and $\hat{\lambda}_{on}$ would be 100 W since these represent the expected value of each state's mean power. In addition, $\hat{r}_{off}$ and $\hat{r}_{on}$ represent the precision in the mean values between different appliance instances, and therefore $\hat{r}_{off}$ would be large since all refrigerators consume close to 0 W when they are off, while $\hat{r}_{on}$ would be relatively

low since the mean on power of different refrigerator instances varies between approximately 80 W and 250 W. Since the precision parameter, $\boldsymbol{\tau}$, varies greatly for different states and appliance instances, the hyperparameters $\hat{\boldsymbol{\beta}}$ and $\hat{\mathbf{w}}$ are used to provide fairly broad prior distribution.

We use this Bayesian approach to parameter estimation in HMMs to individually learn the parameters, $\boldsymbol{\theta}^{(n)}$, of each appliance instance, n, from sequences of their power data, $\mathbf{x}^{(n)}$. Since there is no analytical solution to parameter estimation in HMMs, we performed inference using variational message passing [39], full details of which are included in Appendix A. Variational message passing was used since it provides an efficient and deterministic method of Bayesian parameter estimation for which convergence is guaranteed [27]. We implemented the model as described in this section and performed inference using the Infer.NET framework [27]. In the following section, we describe how these parameters are combined to form a model of the appliance type which will generalise to previously unseen instances of this appliance type.

2.2. Generalising Over Multiple Appliance Instances

We now describe a method by which the parameters learned in Section 2.1 can be combined to form a model that represents the whole appliance type, and therefore generalises to previously unseen instances of that appliance type. Our method consists of fitting distributions to samples drawn from the posterior distributions over appliance instance parameters. As a result, this method averages over our uncertainty around the appliance instance parameters. We introduce the notation:

$$\boldsymbol{\Theta} = \{\boldsymbol{\Theta}^{\alpha}, \boldsymbol{\Theta}^{C}, \boldsymbol{\Theta}^{\lambda}, \boldsymbol{\Theta}^{r}, \boldsymbol{\Theta}^{\beta}, \boldsymbol{\Theta}^{w}\} \tag{9}$$

to represent the parameters of the general model of an appliance type as defined in the following paragraphs. In the case of the refrigerator, $\boldsymbol{\Theta}$ represents a distribution over all possible refrigerator instances. Crucially, this general model allows the probability to be calculated that an appliance instance belongs to the refrigerator appliance type.

Samples drawn from the posterior distributions over the initial probabilities and transition matrix are in the form of multinomial distributions, for which the Dirichlet distribution is the conjugate prior. Therefore, we generalise by fitting Dirichlet distributions to the samples using:

$$\boldsymbol{\Theta}^{\alpha} = \arg\max_{\boldsymbol{\alpha}} \mathrm{Dir}(\boldsymbol{\pi}_{1:M}^{(1:N)} | K, \boldsymbol{\alpha}) \tag{10}$$

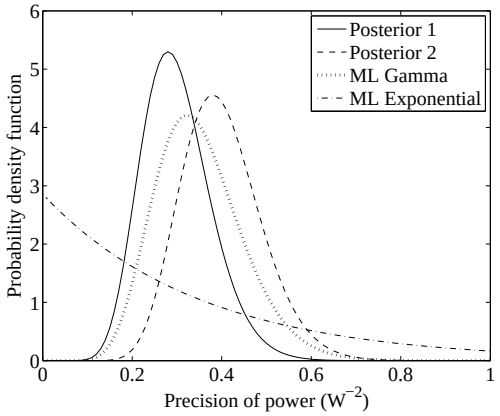


Figure 5: Generalising over the parameters

$$\mathbf{\Theta}_z^C = \arg\max_{\mathbf{C}_z} \mathrm{Dir}(\mathbf{A}_{1:M}^{(1:N)} | K, \mathbf{C}_z) \tag{11}$$

where $\boldsymbol{\pi}_{1:M}^{(1:N)} \sim \mathrm{Dir}(K, \tilde{\boldsymbol{\alpha}})$ and $\mathbf{A}_{1:M}^{(1:N)} \sim \mathrm{Dir}(K, \tilde{\mathbf{C}})$ represent sets of $1, \ldots, M$ samples, and M samples are drawn from the initial and transition posterior distributions for each of the appliance instances $1, \ldots, N$. We use the Fastfit MATLAB toolbox to estimate the parameters of the Dirichlet distributions, which provides a simple and efficient method for parameter estimation through the generalised Newton method [29, 28].

We also fit a Gaussian distribution to samples drawn from the posterior distribution over the emission mean parameters:

$$\mathbf{\Theta}_z^\lambda, \mathbf{\Theta}_z^r = \arg\max_{\lambda, r} \mathcal{N}(\boldsymbol{\mu}_{1:M}^{(1:N)} | \lambda, r) \tag{12}$$

where $\boldsymbol{\mu}_{1:M}^{(1:N)} \sim \mathcal{N}(\tilde{\boldsymbol{\lambda}}, \tilde{\mathbf{r}})$ represents sets of $1, \ldots, M$ samples drawn from the posterior distribution over $1, \ldots, N$ appliance instances' mean power.

Similarly, we fit gamma distributions to samples drawn from the posterior distributions over each state's precision. This results in a distribution which generalises over each posterior distribution of a given state's precision. However, this approach is prone to severe over-fitting when a gamma distribution is fitted to the precisions of the *off* state. In this case, the posterior distributions of the *off* state's precision are often highly peaked and centred around similar values, since they generally only represent the measurement

noise around 0 W. However, it is possible for the power demand to be sampled during a transition between the *off* and *on* states for any appliance type. This results in a sample which would receive near zero probability given the tight estimates of each state's precision. In fact, the probability is likely to be beyond the numerical precision of a double precision floating point number and therefore cause the inference to fail. To prevent this, we constrain the gamma distribution for the *off* state ($k = 1$) which is fitted to samples drawn from the posterior distributions such that it follows an exponential distribution, by holding the shape parameter fixed at 1:

$$\boldsymbol{\Theta}_z^{\beta}, \boldsymbol{\Theta}_z^{w} = \begin{cases} \arg\max_{w} \text{Gamma}(\boldsymbol{\tau}_{1:M}^{(1:N)} | 1, w\}) & k = 1 \\ \arg\max_{\beta, w} \text{Gamma}(\boldsymbol{\tau}_{1:M}^{(1:N)} | \beta, w\}) & k > 1 \end{cases} \tag{13}$$

where $\boldsymbol{\tau}_{1:M}^{(1:N)} \sim \text{Gamma}(\tilde{\boldsymbol{\lambda}}, \tilde{\mathbf{r}})$ represents sets of $1, \ldots, M$ samples drawn from the posterior distribution over $1, \ldots, N$ appliance instances' precision. Figure 5 shows two examples of posterior distributions of the precision of an *off* state as learned from two appliance instances. It can be seen that fitting a gamma distribution to the samples drawn from posterior distributions would result in a tight distribution which would assign an extremely low probability to any measurement of the power demand sampled during a transition between states. Figure 5 also shows an exponential distribution fitted to the samples drawn from the appliance posteriors. It is clear from the long tail shape of this distribution that it will have non-zero support for data points sampled during a transition between two states.

We use the approach described in this section to build models of an appliance type that will generalise to previously unseen instances of that appliance type. We now go on to describe an empirical evaluation of this approach using the Tracebase data set.

2.3. Empirical Evaluation of Model Generalisation using Tracebase Data Set

We evaluated the benefit of building generalisable models of appliance types using the Tracebase data set [33]. This data set is particularly useful for such an evaluation since it contains data from many instances of appliances of the same type. The data set consists of samples of appliances' power demand at roughly one second intervals. We extracted between 2 and 60 signatures (durations when the appliance was in use) depending on data availability for each appliance instance in the Tracebase data set. We selected

Appliance type	Number of instances	Average signatures per instance
Refrigerator	11	19
Kettle	9	14
Microwave	8	8
Washing machine	9	6
Dishwasher	8	19

Table 1: Breakdown of signatures in Tracebase repository

the following 5 common appliance types: refrigerator, kettle, microwave, washing machine and dishwasher. We also extended the Tracebase data set with data collected from 3 additional dishwasher instances, such that at least 8 instances were available for each appliance type. Table 1 shows a breakdown of the signatures extracted from the Tracebase data set.

We modelled the refrigerator, kettle and microwave using the 2 state model shown in Figure 4 (a) and we modelled the washing machine and dishwasher using a 3 state model as shown in Figure 4 (b). We selected the number of states based upon a trade-off between the minimum number of electrical components for each appliance type and the ability to generalise over these states and tune them using aggregate data. For example, refrigerators consist of at least an air compressor which can either be on or off and generate a signature which is clearly visible in the aggregate load, and therefore a 2 state model was appropriate. It is worth noting that although many fridges also contain an interior light, and also fridge-freezers often provide a defrost cycle, such signatures are hard to generalise and almost impossible to extract from the aggregate load, and as a result we chose not to model them. Similarly, a washing machine typically has a water heater and drum motor, both of which generate significant signatures visible in the aggregate load, and therefore a 3 state model was appropriate (including the off state). It is important to note that, although different washing machine cycles are available, the cycles consist of the operation of the same components in different orders. As a result, a 3 state model can represent a range of different cycles. The hyperparameters for each appliance type used are given in Appendix B.

We use hold-one-out cross validation to determine how well a given appliance model generalises to a previously unseen appliance. This involves building a generalisable appliance model using between 2 and 7 training appliance instances, which we show to be sufficient to build a general model

Approach	Description
GT	General appliance model as learned from the Tracebase data set without any parameter tuning.
NT	Specific appliance model as learned from a single appliance instance other than the test appliance.
ST	Specific appliance model as learned from the test appliance instance.

Table 2: Summary of approaches compared using the Tracebase data set.

of each appliance type. We then test these general models against a single appliance instance that was excluded from the training set. Therefore, a single fold of the set of appliance instances corresponds to an ordered list of 7 training appliances and one test appliance. We refold the set of appliance instances 50 times, and for each fold we evaluate how well the appliance model constructed using between 2 and 7 training appliance instances generalises to the test instance.

We compare our approach which builds general models of appliance types (GT) to two bounding benchmarks. The first benchmark uses training data from a single appliance instance from the training set (NT). This represents a lower bound, in which no effort is made to generalise over multiple appliance instances. The second benchmark uses training data from the test appliance (ST). This represents an upper bound, in which the test appliance is not regarded as previously unseen. Since ST and NT are dependent only on the fold of the set of appliance instances, and not on the size of the training set, both values can only be evaluated once for each fold of data. We present the mean log-odds for GT, NT and ST over the 50 folds. These approaches are summarised in Table 2.

We use the model likelihood as a metric for evaluating how well an appliance model explains the test data, averaged over each fold of the data set. This metric represents the likelihood of the test data, $\mathbf{x}$, given a general appliance model, $\boldsymbol{\Theta}$, with both the appliance states $\mathbf{z}$ and parameters $\boldsymbol{\theta}$ integrated out, as given by:

$$p(\mathbf{x}|\boldsymbol{\Theta}) = \iint p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\boldsymbol{\Theta}) \, \mathrm{d}\mathbf{z} \, \mathrm{d}\boldsymbol{\theta} \tag{14}$$

where $p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})$ is calculated using Equation 4 via variational message passing and the Infer.NET framework. Since this likelihood decreases towards zero

as the length of the input data sequence increases, we compare the log-odds rather than the probability. Log-odds, or the logit function, has the advantage that it maps a probability, p, in the range $[0, 1]$ to the domain of real numbers, and therefore avoids problems of numerical precision. This function is defined by:

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right) \tag{15}$$

Figure 6 shows the cross validation model log-odds for 5 common household appliances for training set sizes of between 2 and 7 appliance instances. These are compared with the two benchmarks described above, representing approaches where sub-metered data is available from the test appliance, and where data is only available from a single appliance from the training set. The error bars represent the standard error in the mean. A clear trend common to all appliance types is that the model log-odds increases towards an asymptote as the number of appliance instances in the training set increases. This indicates that the majority of the appliance type's behaviour can be described by a general model learned from a relatively small number of appliance instances. As such, we argue that it is not necessary to build an exhaustive database of all appliance instances as other work has discussed [25, 24], and instead we propose the use of a database of distributions over possible appliance behaviour.

In addition, all averages lie above the lower bounding benchmark, reflecting the intuition that an approach is always preferable if it generalises over multiple appliances rather than uses data from a single instance. Furthermore, all averages lie below the upper bounding benchmark, reflecting that no general model provides a better explanation of sub-metered data than a model learned from that sub-metered data.

Figure 7 shows the normalised cross validation average log-odds for the same 5 appliances. The appliance averages were normalised to lie in the range $[0, 1]$, such that 0 represents the accuracy of the model trained with a single non-test appliance and 1 represents the accuracy of the model trained with the test appliance. This figure enables interesting comparisons between appliance types. First, it can be seen that some appliance types converge towards their asymptote more rapidly than others. This trend is most obvious when comparing the kettle to the washing machine, since the kettle almost converges with a training set of only 3 instances due to its single heating component, while the the washing machine does not converge until

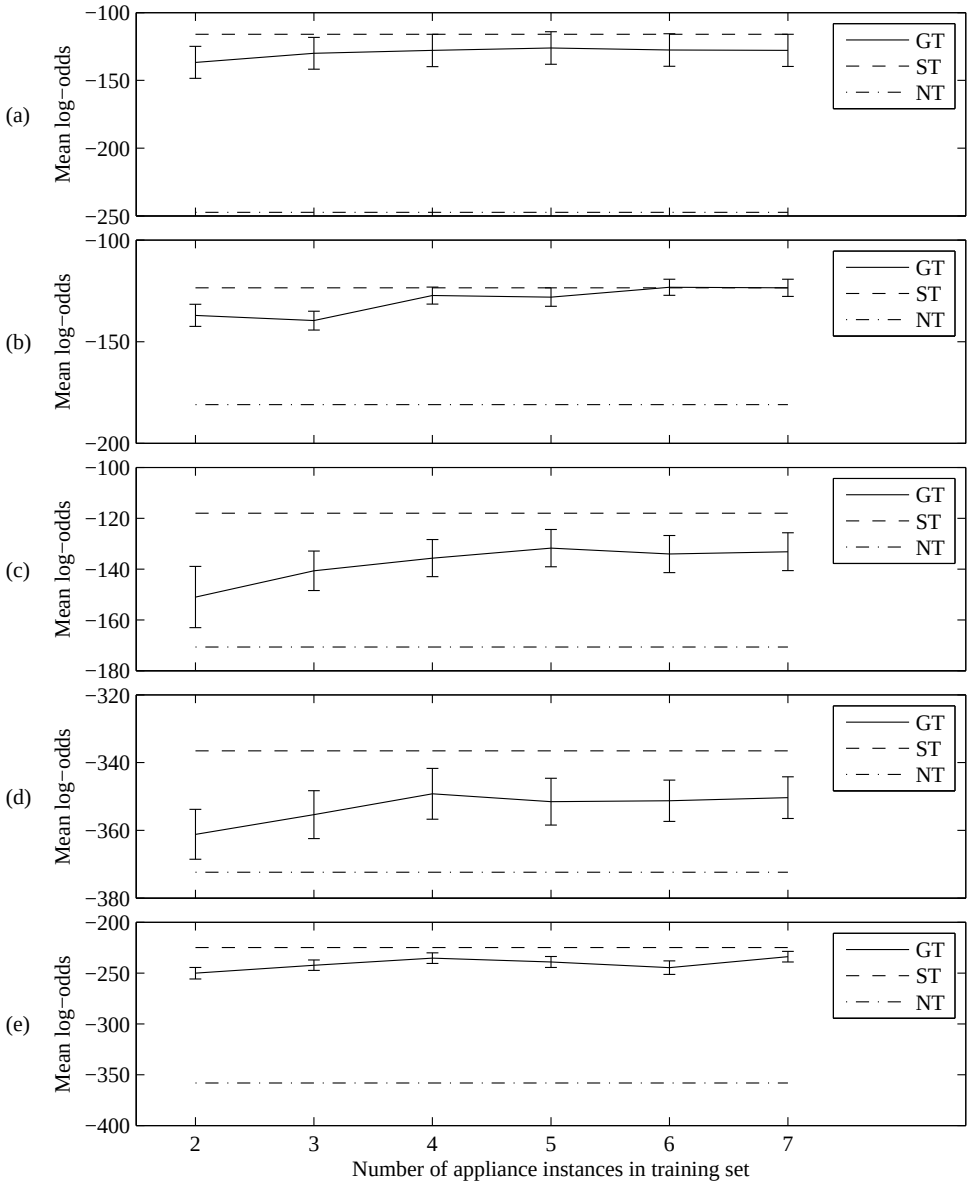


Figure 6: Mean cross validation model log-odds for increasing training set sizes. Legend: GT - generalised training, ST - sub-metered training, NT - non-generalised training. Subplots: (a) Kettle, (b) Refrigerator, (c) Microwave, (d) Washing machine, (e) Dishwasher. Error bars represent standard error in the mean.

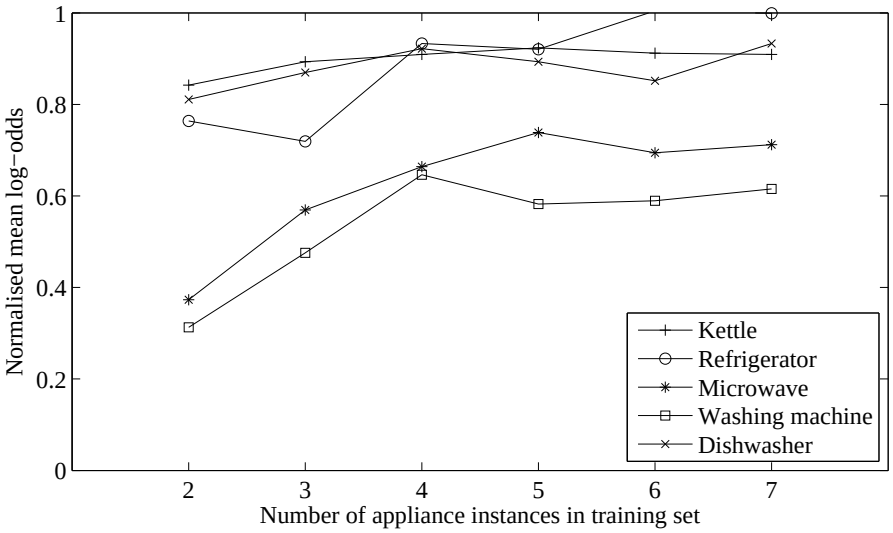


Figure 7: Normalised cross validation model log-odds for increasing training set sizes.

the training set contains 6 appliance instances as a result of its multiple motors and heating elements. This indicates that fewer training examples are required for appliances with fewer electrical components before the optimal general model is achieved. Furthermore, it can be seen that some appliances converge to an asymptote that is closer to the benchmark which uses sub-metered training data. Again, this trend is most obvious when comparing the kettle to the washing machine, since the kettle converges to an asymptote very close to the benchmark which uses sub-metered training data, while the washing machine converges to an asymptote noticeably lower than the corresponding benchmark. This is caused by different degrees of variance within an appliance type, for example, there is less variance within the kettle appliance type than the washing machine appliance type.

Now, having introduced a Bayesian method for inferring the behaviour of appliance instances given a HMM representation, and proposed a novel method for generalising over the multiple appliance instances, we now go on to propose a novel method by which these generalisable appliance models can be tuned to the appliance instances in a new household using only aggregate data.

3. Tuning General Models using Aggregate Data

As identified in the introduction, some appliance behaviour is unique to a particular household and therefore cannot be captured by the general model of the appliance. Such behaviour can be due to the unique characteristics of the appliance instances present in a household (e.g. a freezer with a defrost cycle), and also due to their pattern of usage by the household's occupants (e.g. a microwave often used on low power). Therefore, in this section we propose a method for learning such behaviour that is unique to a single household, which uses both general appliance models and household aggregate data. More formally, this tuning process directly corresponds to learning the parameters $\boldsymbol{\theta}^{(n)}$ for an appliance instance n in a household, given the appliance type's general model $\boldsymbol{\Theta}$ and the household's aggregate data $\mathbf{x}$. Our approach differs from the training approach used by Kim et al. [20], in which appliances are detected using a factorial HMM but are also required to be manually labelled. Similarly, Kolter and Jaakkola [22] proposed a training approach in which individual appliance motifs are extracted from aggregated data, but again each motif was also required to be manual labelled with an appliance name.

Our training approach exploits periods during which a single appliance turns on and off without any other appliances changing state. Such behaviour produces a signature in the aggregate load which is unaffected by all other appliances apart from the base-load. It is these periods which our algorithm uses to tune the general appliance models to specific appliance instances. In the following sections, we first describe how the periods when only a single appliance is operating can be automatically extracted (Section 3.1). We then describe how these periods can be used to tune a general appliance model to the specific appliance within a given a household (Section 3.2). Last, we provide an evaluation using the REDD data set which demonstrates the benefit of model tuning from aggregate data (Section 3.3).

3.1. Extracting Appliance Signatures from an Aggregate Load

As discussed above, our proposed approach requires periods during which a single given appliance is operating to be extracted from an aggregate load. This is achieved by calculating the likelihood that a period of aggregate data was generated by a single appliance instance drawn from a given general appliance model. However, it is important to note that our approach aims to extract periods during which only the appliance of interest is changing state,

Appliance type	Window length (mins)	Sample interval (mins)
Refrigerator	200	5
Microwave	10	1
Washing machine	60	4
Dishwasher	120	4

Table 3: Window length for various appliance types.

and that other appliances might be drawing a constant power during this period. Therefore, in our approach, the base-load is first subtracted from the aggregate load before calculating the likelihood:

$$\bar{\mathbf{x}}_{i:j} = \mathbf{x}_{i:j} - \min(\mathbf{x}_{i:j}) \tag{16}$$

where $\mathbf{x}_{i:j}$ is a window of aggregate data $\mathbf{x}_i, \ldots, \mathbf{x}_j$, and $\bar{\mathbf{x}}_{i:j}$ is the same window after the base-load has been subtracted. This ensures that the distributions over the mean power demand for each state correspond between different signatures. The approach considers windows of aggregate data, for which the size of the window is determined by the maximum signature length encountered in the training data used in Section 2, as shown by Table 3. Longer window lengths can be used for appliances for which multiple sequential signatures can be extracted (e.g. refrigerator), while shorter window lengths should be assigned to appliances which are likely to be used once for a short period of time (e.g. microwave). We calculate the likelihood that a period of aggregate data was generated by a single appliance instance drawn from a given general appliance model as follows:

$$accept(\bar{\mathbf{x}}_{i:j}) = \begin{cases} true & \text{if } p(\bar{\mathbf{x}}_{i:j}|\boldsymbol{\Theta}) > D \\ false & \text{otherwise.} \end{cases} \tag{17}$$

where $\bar{\mathbf{x}}_{i:j}$ is a window of aggregate data after the base-load has been subtracted, $p(\bar{\mathbf{x}}_{i:j}|\boldsymbol{\Theta})$ is the likelihood of that window of data given the general appliance model $\boldsymbol{\Theta}$ as in Equation 14, and D is an appliance specific likelihood threshold. This threshold is set such that the model will accept windows of data which can be explained by a set of appliance parameters drawn from the given appliance type's general model, and reject any windows of data generated by other appliance types or combinations of appliances. Therefore, this process effectively identifies windows of aggregate data during which only an appliance matching its general model changes state. It is worth

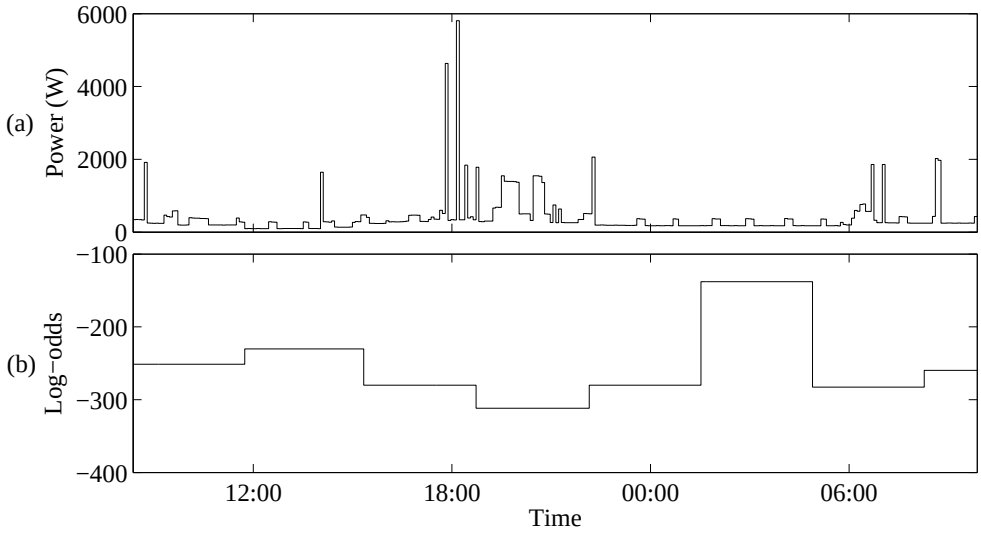


Figure 8: (a) Household power demand over 24 hour period. (b) Log-odds of window of aggregate power being generated by only refrigerator.

noting that the data likelihood $p(\bar{\mathbf{x}}_{i:j}|\boldsymbol{\Theta})$ is inherently dependent upon the variance within an appliance type, since it decreases as the variance of the general appliance model increases. Therefore, we use the sub-metered data from Section 2 to calculate D for each appliance type as the minimum of $p(\mathbf{x}^{(n)}|\boldsymbol{\Theta})$ for each appliance instance n, and as a result this threshold generalises to unseen households.

Figure 8 gives an illustrative example of how appliance signatures can be extracted from an aggregate load in the case of the refrigerator. The figure shows the power demand of a household over a 24 hour period, and also the log-odds that each 4 hour window of data was generated by only the refrigerator. For most windows of data, it is clear that step changes in the aggregate power demand were generated by a combination of the refrigerator and a number of other appliances, and therefore received a low log-odds score. However, between 02:00 and 05:00 only the refrigerator contributed to changes in the aggregate power, and as a result receives a high log-odds score. Therefore, this period can be extracted from the aggregate load and used as an appliance signature with which the general model of refrigerator can be tuned. We found that a step size equal to the window length to be sufficient to extract signatures for each of the modelled appliances. However,

in households where aggregate data is more limited or where overlapping appliance usage is more common, we would expect that a smaller step size would allow a greater number of signatures to be extracted.

3.2. Tuning General Appliance Models using Extracted Signatures

Once the signatures of a single appliance instance have been extracted from aggregate data, the aim is to tune the general model to include the behaviour of the appliance instance which is unique to the previously unseen household. Given that both the general model for this appliance type, $\boldsymbol{\Theta}$, and signatures sampled from the specific appliance instance are available, $\bar{\mathbf{x}}_{i:j}$, Bayesian integration [15] provides a natural approach to infer the posterior distribution over such appliance instance parameters with the state sequence marginalised out:

$$\tilde{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \int p(\bar{\mathbf{x}}_{i:j}, \mathbf{z}|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\boldsymbol{\Theta}) \, \mathrm{d}\mathbf{z} \tag{18}$$

Therefore, we update the general model again using variational message passing as described in Appendix A.

In this setting, Bayesian integration provides a desirable trade-off between parameter tuning and avoiding model over-fitting. For example, when only a small number of appliance signatures are extracted from the aggregate load, the parameters are prevented from becoming over-fitted to one or two signatures. However, when many signatures are extracted from the aggregate load, the parameters are tuned to represent the repeatable behaviour of the appliance instance specific to that household. Since there is no analytical solution to this integral, we again use variational message passing implemented using Infer.NET for the same reasons as in Section 2.1.

Figure 9 illustrates the outcome of the tuning process using the microwave's *on* state as an example. It can be seen that the prior distribution, as learned during the generalisation method described in Section 2, shows a broad distribution over the mean power of all microwaves. In contrast, the posterior distribution, as tuned using the method described in this section, shows a more precise distribution over the mean of this specific microwave instance. However, it should be noted that the mean power of the on state, μ_k, is just one of the set of appliance model parameters, $\boldsymbol{\theta}$, and therefore it is not expected that appliances will be uniquely distinguishable using only this parameter.

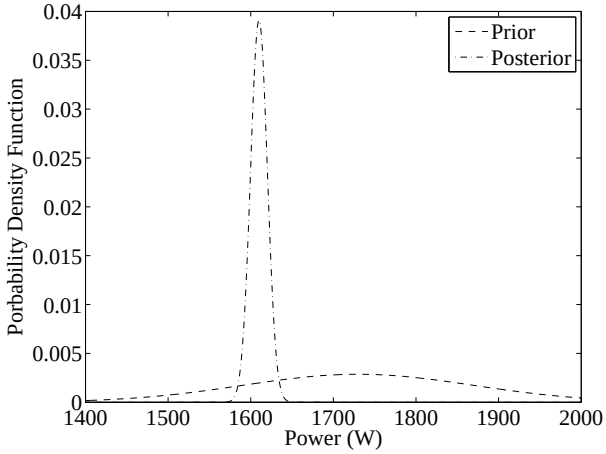


Figure 9: Probability density functions of microwave *on* state mean power prior and posterior distributions.

3.3. Empirical Evaluation of Model Tuning using REDD Data Set

We evaluated the benefit of tuning general appliance models using the Reference Energy Disaggregation Data set (REDD) [23]. This data set was chosen as it is an open data set collected specifically for evaluating NIALM methods, and contains both household aggregate and circuit-level power demand measurements. Since many circuits contain only one appliance, these circuits represent the ground truth power demand for those appliances. As a result, we were able to evaluate how well a given appliance model explains an appliance's actual power demand. However, of the appliances investigated in Section 2, the kettle is not connected to an individual circuit in the REDD data set, and therefore could not be evaluated in this section. Furthermore, due to differences between American and European appliances, it was necessary to artificially increase the mean hyperparameter of the mean power distribution of the on state, $\hat{\lambda}_{on}$, of the general model for the microwave and washing machine. However, all other general model parameters were exactly as learned from the Tracebase data set. We evaluated the appliance models using houses 1-3 from the REDD data set. These homes were selected as they were the only homes which contained at least 1 week of data for each of the refrigerator, microwave, washing machine and dishwasher. However, the washing machine in house 2 was not used throughout period of data collection and therefore was excluded from our experiments. Both aggregate and circuit-level data were down-sampled to a frequency lower than that of mod-

Approach	Description
GT	General appliance model as learned from the Tracebase data set without any parameter tuning.
FT	General appliance model as learned from the Tracebase data set tuned via factorial HMM.
AT	General appliance model as learned from the Tracebase data set tuned using signatures extracted from aggregate data.
ST	General appliance model as learned from the Tracebase data set tuned using signatures extracted from sub-metered data.

Table 4: Summary of approaches compared using REDD data set.

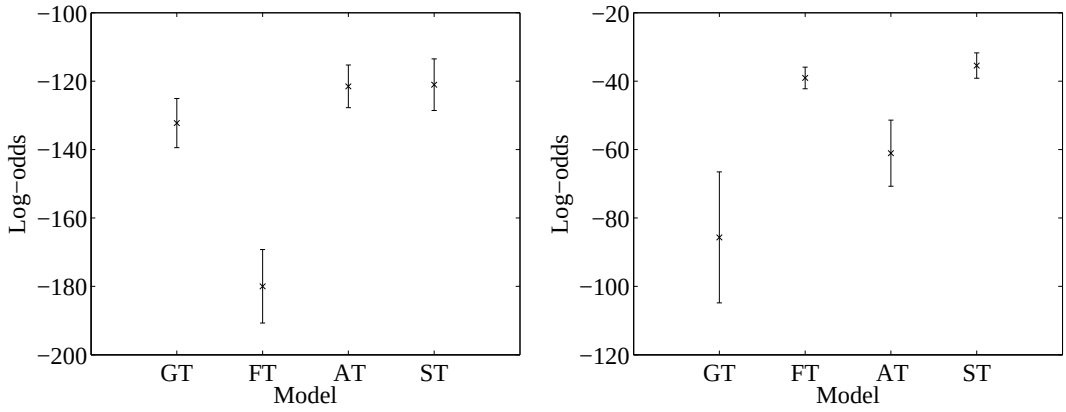
ern smart meters. To ensure realistic computation times, lower sample rates were used for appliances with longer window lengths and higher sample rates were used for appliances with shorter window lengths, as shown by Table 3.

We compare the approach described in this section (AT) to three benchmarks. The first (GT) uses the general appliance model as learned empirically in Section 2 without any model tuning. This variant represents the model fit of the general appliance models. The second benchmark (FT) uses standard Bayesian inference via Gibbs sampling over a factorial HMM when supplied with aggregate data and general appliance priors. The factorial HMM was implemented using the pyhsmm library,[1] in which 4 chains used the same prior as GT, while a further 6 chains used broad priors to capture the behaviour of the unmodelled appliances. This represents the state of the art for unsupervised learning in NIALM [19]. The third benchmark (ST) tunes the general models using sub-metered data, through the approach described in Section 3.2. This approach represents the model fit in the ideal case where sub-metered data is available for model tuning. These four approaches are summarised in Table 4.

As in the previous section, we evaluate the extent to which an appliance model explains the appliance's power demand using the logit function applied to the model likelihood, given by Equation 14 and Equation 15.

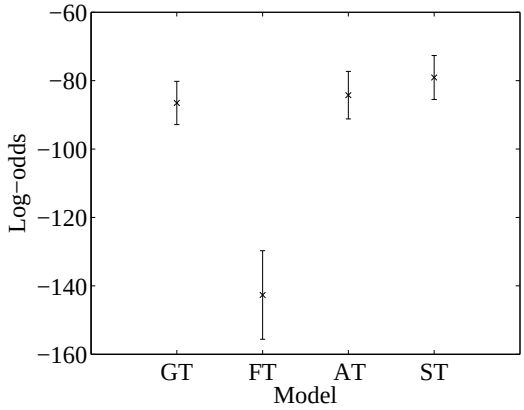
Figure 10 shows the model log-odds for 4 common household appliances averaged over 3 houses, each of which compares the model fit of our proposed approach against the three described benchmarks. The error bars represent

[1] `https://github.com/mattjj/pyhsmm`

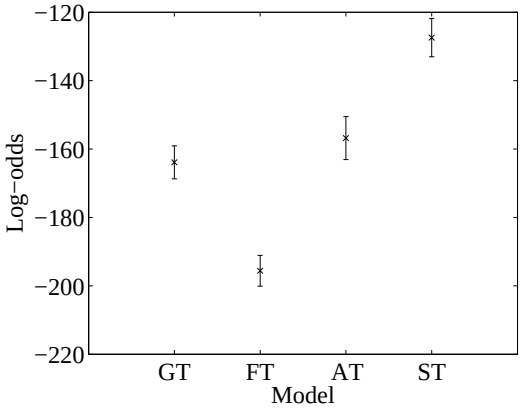


(a) Refrigerator

(b) Microwave



(c) Washing machine



(d) Dishwasher

Figure 10: Mean model log-odds for different training methods. Legend: GT - general model, FT - general model tuned using factorial HMM, AT - general model tuned using extracted signatures, ST - general model tuned using sub-metered data. Error bars represent standard error in the mean.

the standard error in the mean. A clear trend is visible in that the model tuned using signatures extracted from aggregate data (AT) always outperforms the untuned general model (GT). In fact, in the case of the refrigerator it even performs comparably to the model tuned using sub-metered data (ST). This indicates that unsupervised model tuning using aggregate data is a practical alternative to the more intrusive method of supervised training through sub-metered data. Furthermore, it can be seen that for 3 out of 4 appliances, the model tuned using signatures extracted from aggregate data (AT) outperforms the current state of the art (FT) which uses a factorial HMM to tune appliance parameters. This is due to the factorial HMM method often being unable to distinguish between appliances given only general appliance priors and aggregate data, and as a result learns appliance posteriors that consist of combinations of difference appliances.

It is also interesting to compare the benefit of model tuning shown in Figure 10 between appliances. The refrigerator shows the most consistent increase in model log-odds, which can be attributed to the many clean signatures that could be extracted by the AT method with which the general model could be tuned. This is in contrast to the dishwasher, for which fewer, noisier signatures were extracted. As such, there is a smaller increase in the log-odds of the model tuned using aggregate data relative to the model tuned using sub-metered data. This indicates that model tuning will be less effective for appliances often used simultaneously with other appliances. It is also interesting to compare the performance of FT between different appliance types. For the microwave, the FT method is able to improve the general appliance model using only aggregate data, while for the refrigerator, washing machine and dishwasher the FT tuning method actually produces an inferior appliance model.

We now investigate the benefit of tuning each individual parameter of the appliance models. Table 5 shows the Kullback-Leibler (KL) divergence between the models tuned using sub-metered data (ST) and the three approximations (GT, FT, AT) as $D_{\mathrm{KL}}(\mathrm{ST}||\mathrm{GT})$, $D_{\mathrm{KL}}(\mathrm{ST}||\mathrm{FT})$ and $D_{\mathrm{KL}}(\mathrm{ST}||\mathrm{AT})$ respectively, for house 1 of the REDD data set. This table allows the information lost to be compared when each approach is used to approximate the distributions learned from sub-metered data. It can be seen that $D_{\mathrm{KL}}(\mathrm{ST}||\mathrm{FT})$ is systematically greater than $D_{\mathrm{KL}}(\mathrm{ST}||\mathrm{AT})$ for both the transition and emission distributions across all appliances. These high divergence values further highlight that model tuning via signature extraction is preferable to the current state of the art which uses factorial HMMs. It is also interesting to

Appliance	Measure	Initial	Transition	Emission
Refrigerator	$D_{\mathrm{KL}}(\mathrm{ST}\|\|\mathrm{GT})$	0.183	0.164	3.735
	$D_{\mathrm{KL}}(\mathrm{ST}\|\|\mathrm{FT})$	0.369	2.613	26.289
	$D_{\mathrm{KL}}(\mathrm{ST}\|\|\mathrm{AT})$	0.348	0.525	3.376
Microwave	$D_{\mathrm{KL}}(\mathrm{ST}\|\|\mathrm{GT})$	0.072	0.107	1.944
	$D_{\mathrm{KL}}(\mathrm{ST}\|\|\mathrm{FT})$	0.017	1.441	238.544
	$D_{\mathrm{KL}}(\mathrm{ST}\|\|\mathrm{AT})$	1.630	0.469	0.963
Washing machine	$D_{\mathrm{KL}}(\mathrm{ST}\|\|\mathrm{GT})$	0.185	0.057	2.784
	$D_{\mathrm{KL}}(\mathrm{ST}\|\|\mathrm{FT})$	0.008	4.451	5.401
	$D_{\mathrm{KL}}(\mathrm{ST}\|\|\mathrm{AT})$	0.674	0.137	3.599
Dishwasher	$D_{\mathrm{KL}}(\mathrm{ST}\|\|\mathrm{GT})$	0.178	0.356	7.209
	$D_{\mathrm{KL}}(\mathrm{ST}\|\|\mathrm{FT})$	0.599	2.809	9.189
	$D_{\mathrm{KL}}(\mathrm{ST}\|\|\mathrm{AT})$	0.875	0.990	5.079

Table 5: KL divergence between the model tuned using sub-metered data (ST) and the three approximations of this model (GT, FT and AT).

note that although $D_{\mathrm{KL}}(\mathrm{ST}||\mathrm{GT})$ is often slightly less than $D_{\mathrm{KL}}(\mathrm{ST}||\mathrm{AT})$ for the transition matrix, $D_{\mathrm{KL}}(\mathrm{ST}||\mathrm{GT})$ is systematically much greater than $D_{\mathrm{KL}}(\mathrm{ST}||\mathrm{AT})$ for the emission distribution. This indicates that the tuning process is more important for the emission distributions than the transition matrices. However, it is worth noting that the emission distribution of $D_{\mathrm{KL}}(\mathrm{ST}||\mathrm{FT})$ for the microwave received a high divergence score but also explained the actual appliance data with a reasonable likelihood in Figure 10, which indicates that ST is not the only model that can explain actual appliance data with a reasonable likelihood.

We now compare the time taken to tune the general models. Table 6 shows the time taken for the two realistic approaches, FT and AT, to produce the results presented in Figure 10. For the AT method, the vast majority of the time was required to search for windows of aggregate data which could be explained by the general appliance models with a relatively high likelihood, while only a small amount of time was required to update the general models using the extracted signatures. For the FT method, Gibbs sampling was used to approximate the posterior appliance models using a single sequence of aggregate data containing uses of all of the appliances of interest. We found that using 2500 iterations of the Gibbs sampler down-sampled every 25 samples was sufficient to converge towards stationary distributions over the appliance parameters. However, it should be noted that the run-time

House	Time (minutes)	
	AT	FT
1	15.753	32.428
2	10.988	31.788
3	14.946	31.797
Average	13.896	32.005

Table 6: Time taken to tune the general models using periods extracted from aggregate data (AT) and a factorial hidden Markov model (FT).

of FT increases linearly with the number of iterations of the Gibbs sampler, and as such the run-time will vary depending on the number of iterations required.

Having introduced a method by which general appliance models could be tuned to the specific appliances in a household given only aggregate data, and evaluated its performance using the REDD data set, we showed that the tuning process provides an improvement over using general appliance models and the current state-of-the-art tuning method. We now go on to describe the additional benefits that tuned appliance models provide, and give examples of the user feedback that could be produced.

4. Using Tuned Appliance Models to Infer Energy Efficiency

The primary aim of tuning an appliance model is to learn the parameters required to disaggregate that appliance from the household aggregate load. However, such tuned appliance models also provide the required information to derive the specific appliance instance's energy efficiency. For example, the tuned appliance model could be used to calculate the average daily energy consumption of a refrigerator, or the average energy consumption of a washing machine per cycle. These figures can then be converted to compelling feedback that can be provided to the household occupants. Some examples of such feedback are given below:

- **Energy efficiency rating** - An appliance's average energy consumption per day or per use can be mapped to a standard labelling scheme, such as the European Union energy label [7]. This provides the household occupants with an intuitive measure of how their appliance's energy efficiency compares to other similar appliances. Furthermore, it can be compared with the rating quoted by the appliance manufacturer

to determine whether the appliance is operating at its expected level of efficiency.

- **Financial cost** - Annual energy consumption or energy consumption per use can be trivially converted to financial cost using the local cost per kWh of electricity. This provides the occupants with a quantitative measure which can be compared with the cost of other appliances within the household or even a total household bill.

- **Benefit of replacement** - The financial cost of an appliance can also be compared to market leading energy efficient appliances. For continuously operating appliances, such as refrigerators and freezers, the yearly savings can be calculated should the appliance be replaced with a new one. For manually operated appliances, such as washing machines and dishwashers, the per usage savings can be calculated for the replacement appliance. This provides the household occupants with actionable feedback which can be used to decide whether such energy saving actions are worthwhile.

We now give an example of such feedback using the refrigerator from house 1 of the REDD data set. The following calculations are specific to refrigerators and freezers, although this approach could also be adapted to appliances such as washing machines and dishwashers through per usage estimates rather than annual estimates. The tuned refrigerator model can be used to infer the appliance's annual energy consumption (kWh) by weighting each state's mean power demand by the expected time in each state:

$$\mathrm{AC} = \frac{24 \times 365}{1000} \sum_{k=1}^{K} \mu_k \frac{A_{k,k}}{\mathrm{trace}(\mathbf{A})} \tag{19}$$

where trace($\mathbf{A}$) is a function which returns the sum of diagonal elements of matrix $\mathbf{A}$. The European Union energy efficiency index, I, for this refrigerator can then be calculated by comparing the energy consumption against the energy consumption of a standard fridge-freezer of average dimensions, SC:

$$\mathrm{I} = \frac{AC}{SC} \times 100 \tag{20}$$

For this refrigerator, $I = 63.1$, and therefore can be classed as a B band appliance, following the European Union energy efficiency scale [6]. Since this

is the same band as the general model as learned from the Tracebase data set, the household occupants can infer that their refrigerator is of average energy efficiency.

The annual cost of running the refrigerator, C_A, can also be calculated by multiplying the annual consumption by the cost per kWh, C_U:

$$C_A = AC.C_U \tag{21}$$

The annual cost of running this appliance would be £58.08, assuming C_U = £0.15. The household occupants could use this measure to understand the proportional cost of this appliance in comparison to their annual electricity bill.

Last, the benefit of replacing the refrigerator with a highly energy efficient appliance can be calculated. The annual cost of the current refrigerator, C_A, can be compared to that of the market leading energy efficient appliance, C_A^*. Furthermore, the time, T_R, until the annual savings have offset the cost of replacement, C_R, can also be calculated:

$$T_R = \frac{C_R}{C_A - C_A^*} \tag{22}$$

In the case of the REDD house 1 refrigerator, it would take 12 years for a replacement to offset the cost of purchase, assuming C_A^* = £28.82 and C_R = £349. Since this is roughly equal to the average working life of a refrigerator, the household occupants can be advised against replacement. However, if the system were to run continuously the system could alert the household occupants if the energy efficiency decreases such that replacing the appliance is financially viable. Furthermore, in the case that an older F rated refrigerator consuming 920 kWh per year had been in use, the cost of replacement would be offset after only 3 years, and therefore the occupants could be advised to replace the appliance.

5. Conclusions and Future Work

In this paper, we described an unsupervised training method for energy disaggregation systems which tunes general appliance models using only household aggregate data. Unlike existing work, it does not require sub-metered training data from the houses in which disaggregation is performed, nor does it require a manual labelling phase or an exhaustive appliance database. Instead, our approach learns general models of appliance types which generalise

to new appliances, and uses aggregate data as obtained from a smart meter to further learn the characteristics unique to a specific appliance instance.

The proposed generalisation method uses the existing Tracebase appliance data set to learn a HMM for a number of instances of the same appliance type. It then fits distributions to the HMM parameters of each appliance instance to build a general model for each appliance type. Through a cross validation evaluation, we showed that only a small number of instances of each type are required for simple appliances, such as the kettle. However, we also showed that the number of instances required to sufficiently generalise over an appliance type increases for more complex appliances, such as the washing machine.

We also developed a method by which general appliance models can be tuned to the specific appliance instances in a household given only aggregate data as would be collected by a smart meter. The method uses these general models to identify and extract power signatures of individual appliances from the aggregate load, which are subsequently used to tune the general appliance models. We showed that appliance models built using this tuning method explain sub-metered data with a higher likelihood than had the general models been used, and in cases where sufficient signatures can be extracted, such tuned models perform comparably to had sub-metered data been used to tune the models. Furthermore, we showed that our approach outperforms the state of the art, which uses a factorial HMM to tune the appliance parameters using only aggregate data.

The primary motivation for proposing an unsupervised training method for energy disaggregation is to enable existing HMM-based disaggregation methods [20, 22] to be applied automatically to previously unseen households. In addition, we also described the direct benefits which tuned appliance models can provide, such as advice regarding appliance energy efficiency and annual costs.

Future work will focus on a large-scale deployment of the technology presented in this work integrated with AgentSwitch; an agent-based platform designed to help household occupants manage their electricity consumption [32]. We aim to use the general models as constructed from the Tracebase data set, in combination with household aggregate data, to provide intuitive and actionable energy saving advice to household occupants. The accuracy of inferred energy efficiency will be evaluated using limited individual appliance sub-meters, and the operating energy efficiency of appliances will be compared with that quoted by the appliance manufacturer. Furthermore,

we will also use such appliance sub-meters to measure whether the energy saving advice has resulted in energy and financial savings.

In such a deployment, it might be necessary to construct more extensive general appliance models as larger sub-metered data sets become available. However, the use of longer power sequences will increase the time required to build such general models, and therefore more efficient inference algorithms would be required. One possibility would be to exploit the structure of HMMs through a structured variational inference algorithm. Such an approach would iterate between exact inference over HMM states using the Viterbi algorithm [37], and a variational approximation for the HMM parameters.

In most cases, it is trivial to determine the number of states for an appliance type given some examples of power data. However, for more unusual appliances, this might not be the case, and as a result an automated approach will be required to determine the number of states. We believe the infinite hidden Markov model [3] provides a natural representation of appliances in which the number of states is unbounded, and is free to grow as more data is observed. However, new methods will be required to generalise over these models, since an infinite HMM will likely learn a slightly different set of states when applied to multiple appliance instances of the same type.

Another interesting challenge for extending this work would be to apply our proposed training methods to appliance models other than those based on HMMs. We have shown HMMs to be a good model for appliances with discrete set of states (e.g. refrigerator or dishwasher), however HMMs are likely to fail for appliances with a continuously variable power demand (e.g. plasma television). In such cases, different graphical models might represent such continuously variable appliances more appropriately. We believe that the approach proposed in this work, which constructs general appliance models and tunes such models using aggregate data, is general and will be applicable to new graphical models.

Although the majority of UK households contain a combined refrigerator and freezer [41], some households instead contain a separate refrigerator and freezer. In such cases, there might always be at least two appliances which have recently changed state, and therefore it might not be possible to extract individual appliance signatures in order to tune model parameters. As a result, alternative techniques are required which can tune a small number of appliances simultaneously, while still using the generic models to identify such signature periods.

Acknowledgements

The work in this paper was carried out as part of the ORCHID project (EPSRC reference EP/I011587/1).

Appendix A. Approximate Bayesian Inference in hidden Markov models via Variational Message Passing

This appendix first provides a brief overview of variational message passing (VMP), along with the requirements it places upon the Bayesian network structure. We then describe how it can be applied to hidden Markov models, and show how such models fulfil the requirements of VMP. Last, we give an example of the messages that would be passed for a given variable in the Bayesian HMM. Further details of the generality VMP can be found in [39].

VMP is a generalisation of variational inference, which allows variational inference to be applied to arbitrary Bayesian networks. The core advantage of VMP over variational inference is that VMP does not require update equations to be derived manually for each variable in the network. Instead, closed form update equations for each variable can be obtained via the message passing scheme. Messages are passed between variables along the edges of the Bayesian network, and variables are updated upon receiving messages from all variables within its Markov blanket.

However, in order to ensure that closed form update equations can be derived for each node, VMP places the following two restrictions upon the Bayesian network:

1. **Exponential family distributions:** Each variable in the Bayesian network must follow a distribution belonging to the exponential family.
2. **Conjugate priors:** All links in the network must ensure conjugacy along links of conditional dependence.

The Bayesian network we consider in Section 2 and Section 3 consists of a hidden Markov model in which prior distributions are placed over the model parameters. VMP considers both the model parameters and states of the HMM as latent variables, as shown in Figure A.11. In the figure, discrete variables are represented by squares, continuous variables are represented by circles, and fixed hyperparameters are in neither squares nor circles.

In our HMM, the variables follow either Dirichlet, multinomial, Gaussian or gamma distributions. Since all these distributions belong to the expo-

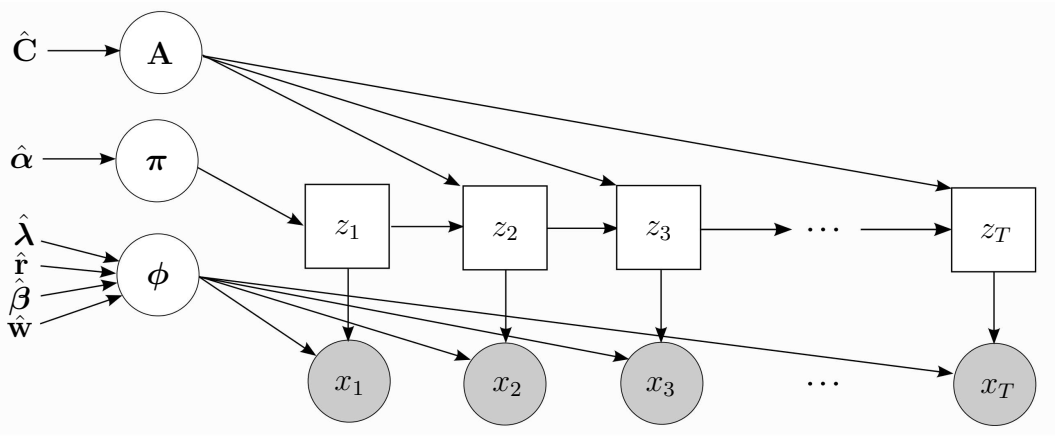


Figure A.11: Bayesian hidden Markov model.

nential family, VMP Requirement 1 is satisfied. Furthermore, the distributions over the model parameters consist of Dirichlet priors over multinomial variables, and Gaussian-gamma priors over Gaussian variables. Since these distributions all represent conjugate prior distributions, VMP Requirement 2 is also satisfied.

Having shown that the graphical structure of Bayesian HMMs respects the requirements of VMP, we now describe how VMP can be used to find local optima in the approximate joint likelihood. We first describe the general message passing algorithm and variable update equations, followed by an example of their application to Bayesian HMMs.

The variational message passing algorithm is defined as follows. First, the variational distribution of each variable is initialised. Next, each variable receives messages from each of its parents and children, before updating its variational distribution. Once all variables have updated their variational distribution, the lower bound on the model likelihood is calculated and the number of iterations is incremented. This process is repeated until either the increase in the lower bound is less than some threshold or the maximum number of iterations has been reached. The algorithm is formalised in Algorithm 1.

We now describe the messages that are passed between variables, and how these messages are used to update each variable's variational distribution. Variational message passing generalises variational inference methods by expressing all distributions over variables in the common form of expo-

Algorithm 1: Variational message passing.

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

nential family distributions:

$$P(\mathbf{X}|\mathbf{Y}) = \exp[\boldsymbol{\phi}(\mathbf{Y})^T \mathbf{u}(\mathbf{X}) + f(\mathbf{X}) + g(\mathbf{Y})] \tag{A.1}$$

where the variable $\mathbf{Y}$ is a parent of variable $\mathbf{X}$, $\boldsymbol{\phi}(\mathbf{Y})$ is the natural parameter vector, $\mathbf{u}(\mathbf{X})$ is the natural statistic vector, and $f(\mathbf{X})$ and $g(\mathbf{Y})$ are other functions specific to the variable's distribution. The messages passed from parents to children are specified by:

$$\mathbf{m}_{Y \to X} = \langle \mathbf{u}_Y \rangle \tag{A.2}$$

where $\langle \mathbf{u}_Y \rangle$ is the expectation of variable Y's natural statistic vector. The messages passed from children to parents are specified by:

$$\mathbf{m}_{X \to Y} = \tilde{\boldsymbol{\phi}}_{XY}(\langle \mathbf{u}_X \rangle, \{\mathbf{m}_{i \to X}\}_{i \in \mathrm{cp}(Y)}) \tag{A.3}$$

where the function $\mathrm{cp}(Y)$ returns the co-parents of variable Y, and $\tilde{\boldsymbol{\phi}}$ is a reparameterisation of $\boldsymbol{\phi}$ in terms of the expectation over the natural statistic vector.

Once messages are received from all parents and children of a variable, the variational distribution over that variable can be updated. The natural parameter vector can be updated by:

$$\boldsymbol{\phi}_Y^* = \tilde{\boldsymbol{\phi}}_Y(\{\mathbf{m}_{i \to X}\}_{i \in \mathrm{pa}(Y)}) + \sum_{j \in \mathrm{ch}(Y)} \mathbf{m}_{j \to Y} \tag{A.4}$$

where the function pa(Y) returns the parents of variable Y and ch(Y) returns the children of variable Y. The natural statistic vector can then be updated by:

$$\langle \mathbf{u}_Y \rangle_{P(\mathbf{X}|\boldsymbol{\phi})} = -\frac{d\tilde{g}(\boldsymbol{\phi})}{d\boldsymbol{\phi}} \tag{A.5}$$

where $\tilde{g}$ is a reparameterisation of g in terms of $\boldsymbol{\phi}$. With each iteration of the VMP algorithm, all variables will update their natural parameter vector via Equation A.4 and natural statistic vector via Equation A.5, and therefore each iteration provides an increase of the lower bound on the model's joint likelihood.

Having provided a general definition of the variational message passing algorithm, we now give an example of the messages that would be received by variable z_2 within the Bayesian HMM in Figure A.11. In order to update variable z_2's variational distribution, it must first receive messages from its parents, z_1 and $\mathbf{A}$, and its children, z_3 and x_2. However, its children will first require messages to be received from its co-parents with x_2, which correspond to variables $\mathbf{A}$ and $\boldsymbol{\phi}$, before they can send their respective messages to z_2. Once z_2 has received messages from all its parents and children, it can then update its variational distribution using Equation A.4 and Equation A.5.

Appendix B. Prior Distributions of Appliance Model Parameters

This section provides the hyperparameter values used for the experiments in Section 2. We used uninformative uniform priors for both Dirichlet distributions over the initial multinomial distribution and transition matrix. We also used rough hyperparameters for the Gaussian-gamma distribution over the emission function, as stated in Table B.7.

References

[1] Armel, K. C., Gupta, A., Shrimali, G., Albert, A., 2013. Is disaggregation the holy grail of energy efficiency? The case of electricity. Energy Policy 52, 213–234.

[2] Baranski, M., Voss, J., 2004. Genetic algorithm for pattern detection in NIALM systems. In: IEEE International Conference on Systems, Man and Cybernetics. Vol. 4. The Hague, Netherlands, pp. 3462 – 3468.

Appliance	Hyperparameter	State		
		1	**2**	**3**
Kettle	$\hat{\boldsymbol{\lambda}}$	0	2000	
	$\hat{\mathbf{r}}$	10^{-2}	10^{-4}	
	$\hat{\boldsymbol{\beta}}$	0.2285	4	
	$\hat{\mathbf{w}}$	0.0088	0.01	
Refrigerator	$\hat{\boldsymbol{\lambda}}$	0	100	
	$\hat{\mathbf{r}}$	10^{-2}	10^{-5}	
	$\hat{\boldsymbol{\beta}}$	0.2285	4	
	$\hat{\mathbf{w}}$	0.0088	0.01	
Microwave	$\hat{\boldsymbol{\lambda}}$	0	1350	
	$\hat{\mathbf{r}}$	10^{-2}	10^{-5}	
	$\hat{\boldsymbol{\beta}}$	0.2285	4	
	$\hat{\mathbf{w}}$	0.0088	0.01	
Washing machine	$\hat{\boldsymbol{\lambda}}$	0	150	1350
	$\hat{\mathbf{r}}$	10^{-2}	10^{-3}	10^{-5}
	$\hat{\boldsymbol{\beta}}$	0.2285	4	4
	$\hat{\mathbf{w}}$	0.0088	0.01	0.01
Dishwasher	$\hat{\boldsymbol{\lambda}}$	0	75	1350
	$\hat{\mathbf{r}}$	10^{-2}	10^{-3}	10^{-5}
	$\hat{\boldsymbol{\beta}}$	0.2285	4	4
	$\hat{\mathbf{w}}$	0.0088	0.01	0.01

Table B.7: Hyperparameters of emission function.

[3] Beal, M. J., Ghahramani, Z., Rasmussen, C. E., 2001. The Infinite Hidden Markov Model. In: Neural Information Processing Systems. Vancouver, BC, Canada.

[4] Berges, M. E., Goldman, E., Matthews, H. S., Soibelman, L., 2010. Enhancing Electricity Audits in Residential Buildings with Nonintrusive Load Monitoring. Journal of Industrial Ecology 14 (5), 844–858.

[5] Chang, H.-H., Lin, C.-L., Lee, J.-K., Apr. 2010. Load identification in nonintrusive load monitoring using steady-state and turn-on transient energy algorithms. In: International Conference on Computer Supported Cooperative Work in Design. Savannah, GA, USA, pp. 27–32.

[6] Council of the European Communities, 1994. COMMISSION DIRECTIVE 94/2/EC of 21 January 1994 implementing Council Directive 92/75/EEC with regard to energy labelling of household electric refrigerators, freezers and their combinations. Official Journal of the European Communities.

[7] Council of the European Communities, 2003. COMMISSION DIRECTIVE 2003/66/EC of 3 July 2003 amending Directive 94/2/EC implementing Council Directive 92/75/EEC with regard to energy labelling of household electric refrigerators, freezers and their combinations. Official Journal of the European Communities.

[8] Darby, S., 2006. The Effectiveness of Feedback on Energy Consumption, A review for DEFRA of the literature on metering, billing and direct displays. Tech. rep., University of Oxford, UK.

[9] Department of Energy & Climate Change, 2008. Climate Change Act. Tech. rep., UK.

[10] Department of Energy & Climate Change, 2009. A consultation on smart metering for electricity and gas. Tech. rep., UK.

[11] Department of Energy & Climate Change, 2013. Smart Metering Equipment Technical Specifications Version 2. Tech. rep., UK.

[12] Ehrhardt-Martinez, K., Donnelly, K. A., Laitner, J. A. S., 2010. Advanced Metering Initiatives and Residential Feedback Programs: A Meta-Review for Household Electricity-Saving Opportunities. Tech.

rep., Research Report E105, American Council for an Energy-Efficient Economy.

[13] Farinaccio, L., Zmeureanu, R., 1999. Using a pattern recognition approach to disaggregate the total electricity consumption in a house into the major end-uses. Energy and Buildings 30 (3), 245–259.

[14] Froehlich, J., Larson, E., Gupta, S., Cohn, G., Reynolds, M., Patel, S., 2011. Disaggregated End-Use Energy Sensing for the Smart Grid. Pervasive Computing, IEEE 10 (1), 28–39.

[15] Ghahramani, Z., 2001. An Introduction to Hidden Markov models and Bayesian Networks. In: Journal of Pattern Recognition and Artificial Intelligence. Vol. 15. pp. 9–42.

[16] Gonçalves, H., Ocneanu, A., Bergés, M., 2011. Unsupervised disaggregation of appliances using aggregated consumption data. In: ACM Special Interest Group on Knowledge Discovery and Data Mining, workshop on Data Mining Applications in Sustainability. San Diego, CA.

[17] Gupta, S., Reynolds, M. S., Patel, S. N., 2010. ElectriSense: single-point sensing using EMI for electrical event detection and classification in the home. In: Proceedings of the 12th ACM International Conference on Ubiquitous Computing. New York, NY, USA, pp. 139–148.

[18] Hart, G. W., 1992. Nonintrusive appliance load monitoring. Proceedings of the IEEE 80 (12), 1870–1891.

[19] Johnson, M. J., Willsky, A. S., 2013. Bayesian Nonparametric Hidden Semi-Markov Models. Journal of Machine Learning Research 14, 673–701.

[20] Kim, H., Marwah, M., Arlitt, M. F., Lyon, G., Han, J., 2011. Unsupervised Disaggregation of Low Frequency Power Measurements. In: Proceedings of the 11th SIAM International Conference on Data Mining. Mesa, AZ, USA, pp. 747–758.

[21] Kolter, J. Z., Batra, S., Ng, A. Y., 2010. Energy Disaggregation via Discriminative Sparse Coding. In: Proceedings of the 24th Annual Conference on Neural Information Processing Systems. Vancouver, BC, Canada, pp. 1153–1161.

[22] Kolter, J. Z., Jaakkola, T., 2012. Approximate Inference in Additive Factorial HMMs with Application to Energy Disaggregation. In: Proceedings of the 15th International Conference on Artificial Intelligence and Statistics. La Palma, Canary Islands, pp. 1472–1482.

[23] Kolter, J. Z., Johnson, M. J., 2011. REDD: A Public Data Set for Energy Disaggregation Research. In: ACM Special Interest Group on Knowledge Discovery and Data Mining, workshop on Data Mining Applications in Sustainability. San Diego, CA, USA.

[24] Lai, P.-h., Trayer, M., Ramakrishna, S., Li, Y., 2012. Database Establishment for Machine Learning in NILM. In: 1st International Workshop on Non-Intrusive Load Monitoring. Pittsburgh, PA, USA.

[25] Lam, H. Y., Fung, G. S. K., Lee, W. K., 2007. A Novel Method to Construct Taxonomy of Electrical Appliances Based on Load Signatures. IEEE Transactions on Consumer Electronics 53 (2), 653–660.

[26] Marceau, M. L., Zmeureanu, R., 2000. Nonintrusive load disaggregation computer program to estimate the energy consumption of major end uses in residential buildings. Energy Conversion and Management 41 (13), 1389–1403.

[27] Minka, T., Winn, J. M., Guiver, J. P., Knowles, D. A., 2012. Infer.NET 2.5. Tech. rep., Microsoft Research Cambridge.

[28] Minka, T. P., 2002. Beyond Newton's method. Tech. rep.

[29] Minka, T. P., 2002. Estimating a Dirichlet distribution. Tech. rep.

[30] Murphy, K. P., 2007. Conjugate Bayesian analysis of the Gaussian distribution. Tech. rep.

[31] Parson, O., Ghosh, S., Weal, M., Rogers, A., 2012. Non-intrusive load monitoring using prior models of general appliance types. In: Proceedings of the 26th AAAI Conference on Artificial Intelligence. Toronto, ON, Canada, pp. 356–362.

[32] Ramchurn, S., Osborne, M., Parson, O., Rahwan, T., Maleki, S., Reece, S., Huynh, T. D., Alam, M., Fischer, J., Rodden, T., Moreau, L.,

Roberts, S., 2013. AgentSwitch: Towards smart electricity tariff selection. In: Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems. Saint Paul, MN, USA.

[33] Reinhardt, A., Bauman, P., Burgstahler, D., Hollick, M., Chonov, H., Werner, M., Steinmetz, R., 2012. On the Accuracy of Appliance Identification Based on Distributed Load Metering Data. In: Proceedings of the 2nd IFIP Conference on Sustainable Internet and ICT for Sustainability. Pisa, Italy, pp. 1–9.

[34] Rogers, A., Ramchurn, S., Jennings, N. R., 2012. Delivering the smart grid: Challenges for autonomous agents and multi-agent systems research. In: Proceedings of the 26th AAAI Conference on Artificial Intelligence. Toronto, ON, Canada, pp. 2166–2172.

[35] Srinivasan, V., Stankovic, J., Whitehouse, K., 2013. FixtureFinder: Discovering the Existence of Electrical and Water Fixtures. In: 12th ACM/IEEE Conference on Information Processing in Sensor Networks. Philadelphia, PA, USA. In press.

[36] U.S. Department of Energy, 2003. Grid 2030: A National Vision For Electricitys Second 100 Years. Tech. rep.

[37] Viterbi, A., 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory 13 (2), 260–269.

[38] Weiss, M., Helfenstein, A., Mattern, F., Staake, T., 2012. Leveraging smart meter data to recognize home appliances. In: Proceedings of the IEEE International Conference on Pervasive Computing and Communications. Lugano, Switzerland, pp. 190–197.

[39] Winn, J., Bishop, C. M., 2006. Variational message passing. Journal of Machine Learning Research 6 (1), 661–694.

[40] Zeifman, M., Roth, K., 2011. Viterbi algorithm with sparse transitions (VAST) for nonintrusive load monitoring. In: IEEE Symposium on Computational Intelligence Applications In Smart Grid. pp. 1–8.

[41] Zimmermann, J.-P., Evans, M., Griggs, J., King, N., Harding, L., Roberts, P., Evans, C., 2012. Household Electricity Survey A study of domestic electrical product usage. Tech. rep., Intertek.