Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence

Non-Intrusive Load Monitoring Using Prior Models of General Appliance Types

Oliver Parson, Siddhartha Ghosh, Mark Weal, Alex Rogers
Electronics and Computer Science
University of Southampton, Hampshire, SO17 1BJ, UK
{*op106,sg2,mjw,acr*}*@ecs.soton.ac.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 approach by which individual appliances can be iteratively separated from an aggregate load. Unlike existing approaches, our approach does not require training data to be collected by sub-metering individual appliances, nor does it assume complete knowledge of the appliances present in the household. Instead, we propose an approach in which prior models of general appliance types are tuned to specific appliance instances using only signatures extracted from the aggregate load. The tuned appliance models are then used to estimate each appliance's load, which is subsequently subtracted from the aggregate load. This process is applied iteratively until all appliances for which prior behaviour models are known have been disaggregated. We evaluate the accuracy of our approach using the REDD data set, and show the disaggregation performance when using our training approach is comparable to when sub-metered training data is used. We also present a deployment of our system as a live application and demonstrate the potential for personalised energy saving feedback.

1 Introduction

Non-intrusive appliance load monitoring (NIALM), or energy disaggregation, aims to break down a household's aggregate electricity consumption into individual appliances (Hart 1992). The motivations for such a process are twofold. First, informing a household's occupants of how much energy each appliance consumes empowers them to take steps towards reducing their energy consumption (Darby 2006). Second, if the NIALM system is able to determine the current time of use of each appliance, a recommender system would be able to inform a household's occupants of the potential savings through deferring appliance use to a time of day when electricity is either cheaper or has a lower carbon footprint. To address these goals through a practical and widely applicable software system, it is essential to take advantage of existing infrastructure rather than designing new hardware. Smart meters are currently being deployed on national scales (Department of Energy & Climate Change 2009) and thus constitute an ideal data collection platform for NIALM solutions.

Recent contributions to the field of NIALM have applied principled machine learning techniques to the problem of energy disaggregation. Such approaches fall into two categories. The first uses supervised methods which assume that sub-metered (ground truth) data is available for training prior to performing disaggregation (Kolter, Batra, and Ng 2010; Kolter and Johnson 2011). This assumption dramatically increases the investment required to set up such a system, since in practice, installing sub-meters may be inconvenient or time consuming. The second uses unsupervised disaggregation methods (Kim et al. 2011; Zeifman and Roth 2011; Kolter and Jaakkola 2012) in which no prior knowledge of the appliances is assumed, but which often require appliances to be manually labelled after the disaggregation process or assume knowledge of the number of household appliances. Such approaches also typically ignore additional information that may be available regarding which appliances are most likely to be present in a house or how such appliances are likely to behave.

While these assumptions are attractive from a machine learning perspective, they do not address the most likely real world applications of NIALM, where sub-metered data and complete knowledge of the appliance set is not available, but some prior information about some appliances might be known. This prior information exists as expert knowledge of an appliance's mode of operation (e.g. its power demand and usage cycle), and can be encoded as a generic appliance model. This information is important as it can be used to automate the process of labelling disaggregated appliances and even be used to identify appliances which unsupervised methods cannot. It is therefore necessary to design training methods that are able to utilise both generic appliance models and aggregate consumption data without requiring complete knowledge about the type and number of all the appliances within the home.

It is exactly this challenge we address in this paper, and to do so, we adopt a graphical representation which incorporates the difference hidden Markov model (HMM) (Kolter and Jaakkola 2012), to disaggregate single appliances from household aggregate power readings. The difference HMM is well-suited to NIALM as it explicitly represents step changes in the aggregate power as observed data. In contrast to Kolter and Jaakkola's unsupervised training method, we describe an approach in which generic appliance models and aggregate consumption data are used to generate models of specific appliance instances using expectation-maximisation (EM). We then describe a method which uses these trained models to disaggregate individual appliances using an extension of the Viterbi algorithm. We focus on disaggregating common appliance types which consume a large proportion of the home's energy, particularly those whose use may be deferred by the household occupants (e.g. washing machine, clothes dryer). We evaluate the accuracy of our proposed approach using the Reference Energy Disaggregation Dataset (REDD) (Kolter and Johnson 2011), before describing a deployment of our NIALM system as a real-time application.

Our contributions are summarised as follows:

- We describe a novel training process in which prior knowledge of the generic appliance types are tuned to specific appliance instances using only aggregate data from the home in which disaggregation is being performed. We represent each appliance using a probabilistic graphical model, and our training process corresponds to learning the parameters of this model. The graphical model along with the learned set of parameters governing the variable distributions constitute a model of an appliance. To learn such parameters, clean signatures of individual appliances are first identified within the aggregate signal by applying the expectation-maximisation algorithm to small overlapping windows of aggregate data. These clean appliance signatures are then used to tune generic models of appliance types to the household's specific appliances.
- We present a novel iterative disaggregation method that models each appliance's load using our graphical model and disaggregates them from the aggregate power demand. Our disaggregation method uses an extension of the Viterbi algorithm (Viterbi 1967) which filters the aggregate signal such that interference from other appliances is ignored. The disaggregated appliance's load is then subtracted from the aggregate load. This process is repeated until all appliances for which general models are available have been disaggregated from the aggregate load.
- We evaluate the accuracy of our proposed approach using the REDD dataset (http://redd.csail.mit.edu/). This dataset contains both the aggregate and circuit-level power demands for a number of US households. Since many appliances operate on their own circuit, we can use this data as ground truth to compare against the output of our disaggregation approach. We down sample the data to 1 minute resolution, since this is typical of the high data rate that an in-home display might receive data from a smart meter. We benchmark against two variants of our approach, and show that the disaggregation performance when using our training approach is comparable to when sub-metered training data is used.
- Finally, we present a deployment of our NIALM system as a live application. We show that our approach is robust against noisy and missing data, as is the case in real deployments. Since sub-metered data is not available in this setting we do not use this deployment to test the accuracy of our approach, but instead demonstrate its ability to infer previously unknown data without sub-metered training.

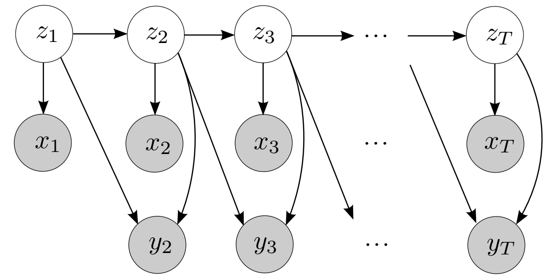


Figure 1: Our difference HMM variant. Shaded nodes represent observed variables and unshaded nodes represent hidden variables.

The remainder of this paper is organised as follows. In Section 2 we formalise the problem of NIALM. Section 3 defines a graphical model of the system and describes how it can be trained and used to solve the NIALM problem. In Section 4 we empirically evaluate our approach using REDD. In Section 5 we describe a deployment of our approach as a live application, and we conclude in Section 6.

2 Problem Description

Formally, the aim of NIALM is as follows. Given a discrete sequence of observed aggregate power readings $\mathbf{x} = x_1, \ldots, x_T$, determine the sequence of appliance power demands $\mathbf{w}^{(n)} = w_1^{(n)}, \ldots, w_T^{(n)}$, where n is one of N appliances. Alternatively, this problem can be represented as the determination of appliances states, $\mathbf{z}^{(n)} = z_1^{(n)}, \ldots, z_T^{(n)}$, if a mapping between states and power demands is known. Each appliance state corresponds to an operation of approximately constant power draw (e.g. 'on', 'off' or 'standby') and t represents one of T discrete time measurements.

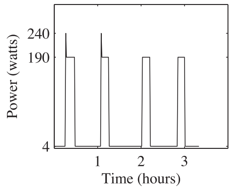
3 Energy Disaggregation Using Iterative Hidden Markov Models

In this section, we describe how we model each appliance and how these generic models of appliance types are trained to specific appliance instances using aggregate data. The fully trained models are then used to disaggregate the appliance's power demand from the aggregate power demand.

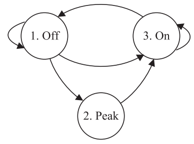
3.1 Appliance Models

Our approach models each appliance as a variant of the difference hidden Markov model (HMM) of Kolter and Jaakkola (2012), where step changes in the aggregate power are modelled explicitly as an observation sequence as shown in Figure 1. In the model each latent discrete variable, z_t, in the Markov chain represents the state of the appliance at an instant in time. Each variable z_t takes on an integer value in the range $[1, K]$ where K is the number of states.

In a standard HMM, each variable in the Markov chain emits a single observation. However, in our model we consider two observation sequences $\mathbf{x}$ and $\mathbf{y}$. Sequence $\mathbf{x}$ corre-



(a) Prior of appliance power.



(b) State transition model.

z_t	μ	σ^2
1	4 W	10 W^2
2	240 W	150 W^2
3	190 W	100 W^2

(c) Emission density ϕ.

z_t	1	2	3
1	0.95	0.03	0.02
2	0	0	1
3	0.1	0	0.9

(d) Transition matrix $\mathbf{A}$.

Figure 2: Refrigerator model parameters

sponds to the household aggregate power demand measured by the smart meter. These aggregate power observations are used to restrict the time slices in which an appliance can be 'on' to only those when the aggregate power demand is greater than that of the individual appliance. Sequence $\mathbf{y}$ is derived from $\mathbf{x}$, and corresponds to the difference between two consecutive aggregate power readings such that $y_t = x_t - x_{t-1}$ (hence this model is referred to as a difference HMM). These derived observations are used to infer the probability that a change in aggregate power, y_t, was generated by two consecutive appliance states.

In a standard HMM, each observed variable is conditionally dependent on a single latent variable. However, in our variant of the difference HMM, $\mathbf{y}$ represents the difference in aggregate power between consecutive time slices, and is therefore dependent on the appliance state in both the current and previous time slice. Figure 1 shows these dependencies.

For each appliance n, the dependencies between the variables in our graphical model can be defined by a set of three parameters: $\boldsymbol{\theta}^{(n)} = \{\boldsymbol{\pi}^{(n)}, \mathbf{A}^{(n)}, \boldsymbol{\phi}^{(n)}\}$, respectively corresponding to the probability of an appliance's initial state, the transition probabilities between states and the probability that an observation was generated by an appliance state. The rest of this section defines the functions these parameters govern, omitting appliance indices $^{(n)}$ for conciseness.

The probability of an appliance's starting state at $t = 1$ is represented by the vector $\boldsymbol{\pi}$ such that:

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

The transition probabilities from state i at $t-1$ to state j at t are represented by the matrix $\mathbf{A}$ such that:

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

We assume that each appliance has a Gaussian distributed power demand:

$$w_t | z_t, \boldsymbol{\phi} \sim \mathcal{N}(\mu_{z_t}, \sigma^2_{z_t}) \tag{3}$$

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

$$y_t | z_t, z_{t-1}, \boldsymbol{\phi} \sim \mathcal{N}(\mu_{z_t} - \mu_{z_{t-1}}, \sigma^2_{z_t} + \sigma^2_{z_{t-1}}) \tag{4}$$

where $\boldsymbol{\phi}_k = \{\mu_k, \sigma^2_k\}$, and μ_k and σ^2_k are the mean and variance of the Gaussian distribution describing this appliance's power draw in state k. Equation 4 is used to evaluate the probability that a change in the aggregate power was generated by an appliance transition between two states.

Equations 1, 2 and 4 are the minimum definitions needed to define a difference HMM. However, by using the change in aggregate power as an observation sequence, the model does not impose the constraint that appliances can only be 'on' when the observed aggregate power is greater than the appliance's power. We impose this constraint by considering the aggregate power demand, x_t, as a censored reading of an appliance's power demand, w_t, which we incorporate into our model using an additional emission function representing the cumulative distribution function of an appliance's Gaussian distributed power demand:

$$\begin{aligned} P(w_{z_t} \leq x_t | z_t, \boldsymbol{\phi}) &= \int_{-\infty}^{x_t} \mathcal{N}(\mu_{z_t}, \sigma^2_{z_t}) \mathrm{d}w \\ &= \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{x_t - \mu_{z_t}}{\sigma_{z_t}\sqrt{2}}\right)\right] \end{aligned} \tag{5}$$

This emission function constrains the model such that when the aggregate power reading is much less than the mean power draw of an appliance state, then the probability of that state will tend towards 0. However, if the aggregate power reading is much greater than the mean power draw of the appliance state, the emission probability of that state will tend towards 1. If it were applied independently to each appliance, this constraint would be a relaxation of the constraint that the sum of the all the appliance's power demands must equal the aggregate power measurement: $\sum_{n=1}^{N} \mu^{(n)}_{z_t} = x_t$. However, our approach subtracts the power demand of an appliance from the aggregate according to $\hat{x}_t = x_t - \mu^{(n)}_{z_t}$ before disaggregating the next appliance. This effectively couples the appliances and therefore ensures the sum of the subset of appliances which we disaggregate is always likely to be less than the observed aggregate power.

The model parameters $\boldsymbol{\theta}$ are learned from aggregate data as described in Section 3.2 and used to disaggregate appliance loads in Section 3.3.

3.2 Training Using Aggregate Data

Our novel training approach takes generic models of appliance types, $\hat{\boldsymbol{\theta}}$ (e.g. all clothes dryers), and trains them to specific appliance instances (e.g. a particular clothes dryer appliance installed in a particular home) using only a household's aggregate power demand. This approach differs from the unsupervised training approaches used by Kim et al.

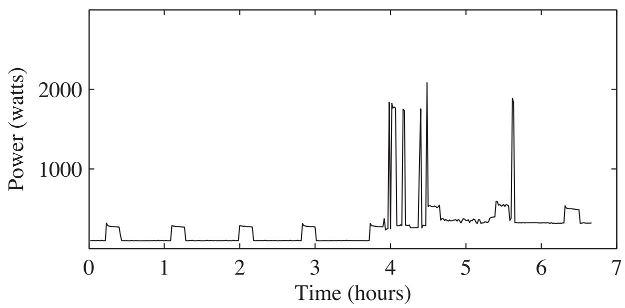


Figure 3: Example of aggregate power demand

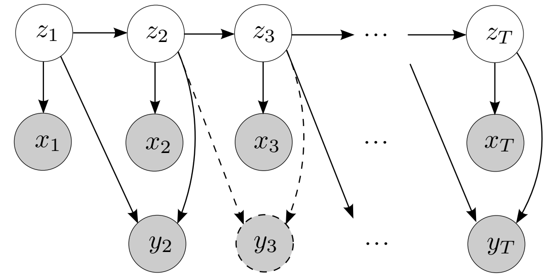


Figure 4: A difference HMM variant where observation y_3 shown by dashed lines has been filtered out.

(2011) and Kolter and Jaakkola (2012), as we use prior knowledge of appliance behaviour and power demands. This prior knowledge includes appliance type labels, and therefore does not require the manual labelling of disaggregated appliances. This training process directly corresponds to learning values for each appliance's model parameters $\boldsymbol{\theta}^{(n)}$.

The generic model of an appliance type consists of priors over each parameter of an appliance's model. The prior state transition matrix consists of a matrix, in which possible transitions between states are represented by a probability between zero and one, and transitions which are not possible in practice are represented by a probability of zero. The prior over an appliance's emission function consists of expected values of the Gaussian distribution's mean and variance. Figure 2 (a) shows how an expert might expect a refrigerator to operate, while (b) shows a corresponding state transition model. It is important to note that the peak power draw is not always captured due to the low sampling rate, and is represented in the transition model by a direct transition between the 'off' and 'on' states and also by an indirect transition via the 'peak' state. From this information, values for the appliance's prior can be inferred as shown in Figures 2 (c) and (d). The emission density parameters govern the distribution of the appliance's power draw and the transition probabilities are proportional to the time spent in each state.

An appliance prior should should be general enough to be able to represent many appliance instances of the same type (e.g. all refrigerators). However, if a small number of distinct behavioural categories exist for a single appliance type, it might be suitable to use more than one prior model for that appliance type (e.g. 'hot' and 'cold' cycle for a washing machine or the 'high' and 'eco' modes for an electric shower as we show in Section 5). These prior models could be determined in a number of ways. Most directly, a domain expert could estimate an appliance's emission density using knowledge of the its power demand available from appliance documentation. Additionally, the transition matrix could be estimated using expert knowledge of the expected duration of each state and the average time between uses. Alternatively, these parameters may be constructed by generalising data collected from laboratory trials or other sub-metered homes.

Our training approach exploits periods during which a single appliance turns on and off without any other appliances changing state. This produces a signature in the aggregate load which is unaffected by all other appliances apart from the baseline load. It is these periods which our algorithm uses to train the appliance models to the specific appliance instances. It is often the case where some appliance's signatures are easier to extract and loads are simpler to disaggregate than others. Therefore, by training and disaggregating each appliance in turn we can gradually clean the aggregate signal, causing the remaining signatures to become more prominent. Figure 3 shows an example of the aggregate power demand. From hours 1 to 3 it is clear that only the refrigerator is cycling on and off. This period can be used to train the refrigerator appliance model, which is then used to disaggregate the refrigerator's load for the whole duration. Subtracting the refrigerator's load will consequently clean the aggregate load allowing additional appliances to be identified and disaggregated.

Our approach to tune such general models to specific appliance instances is as follows. First, data to train an appliance model must be extracted from the aggregate load. This is achieved by running the EM algorithm on small overlapping windows of aggregate data. During training, we use a reduced graphical model containing only sequences $\mathbf{z}$ and $\mathbf{y}$. The EM algorithm is initialised with our prior appliance's state transition matrix and power demand. Our prior state transition matrix is sparse as it contains mostly zeros, therefore restricting the range of behaviours that it can represent (Bishop 2006). The EM algorithm terminates when a local optima in the log likelihood function has been found or a maximum number of iterations has been reached. The function defining the acceptance of windows for training upon termination of the EM algorithm can be described as follows:

$$accept(x_i, \ldots, x_j|\hat{\boldsymbol{\theta}}) = \begin{cases} true & \text{if } \ln \mathcal{L} > D \\ false & \text{otherwise.} \end{cases} \quad (6)$$

where $x_i, \ldots, x_j$ is a window of data, $\mathcal{L}$ is the likelihood of the window of data given the prior over the model parameters $\hat{\boldsymbol{\theta}}$, and D is an appliance specific likelihood threshold. As such, the model will reject windows in which the prior model cannot be tuned to explain the observations with a log likelihood greater than the threshold. Therefore, this process effectively identifies windows of aggregate data during which only one appliance changes state. Next, the accepted data windows are used to tune our prior appliance model $\hat{\boldsymbol{\theta}}$

Appliance	Av. uses	NT	AT	ST
Refrigerator	262	$38\% \pm 4\%$	$21\% \pm 2\%$	$55\% \pm 6\%$
Microwave	73	$63\% \pm 4\%$	$53\% \pm 5\%$	$38\% \pm 3\%$
Clothes dryer	43	$3469\% \pm 492\%$	$55\% \pm 2\%$	$71\% \pm 5\%$
Air conditioning	26	$57\% \pm 1\%$	$77\% \pm 1\%$	$65\% \pm 1\%$

Table 1: Mean normalised error in total assigned energy per day over all houses

Appliance	Av. uses	NT	AT	ST
Refrigerator	262	84 W ± 1 W	77 W ± 1 W	83 W ± 1 W
Microwave	73	131 W ± 2 W	124 W ± 2 W	111 W ± 2 W
Clothes dryer	43	3107 W ± 18 W	422 W ± 7 W	474 W ± 8 W
Air conditioning	26	559 W ± 12 W	477 W ± 11 W	455 W ± 11 W

Table 2: RMS error in assigned power over all time slices

to our fully trained appliance model $\boldsymbol{\theta}$ using a single application of the EM algorithm over multiple data sequences.

3.3 Disaggregation via Extended Viterbi Algorithm

The disaggregation task aims to infer each appliance's load given only the aggregate load and the learned appliance's parameters $\boldsymbol{\theta}^{(n)}$. After learning the parameters for each appliance which we wish to disaggregate, any inference mechanism capable of disaggregating a subset of appliances could be used. We use an extension of the Viterbi algorithm which is able to iteratively disaggregate individual appliances by filtering out observations generated by other appliances and disaggregate the modelled appliance in parallel.

The Viterbi algorithm can be used to determine the optimal sequence of states in a HMM given a sequence of observations. For the Viterbi algorithm to be applicable to our graphical model, it must be robust to other unmodelled appliances contributing noise to the observed load, and must also process our two observation sequences. To this end, we extend the algorithm in two ways.

First, we allow the forward pass to filter out observations for which the joint probability is less than a predefined appliance specific threshold C:

$$t \in \mathbf{S} = \begin{cases} true & \text{if } \max\limits_{z_{t-1}, z_t} (p(y_t, z_{t-1}, z_t|, \boldsymbol{\theta})) \geq C \\ false & \text{otherwise.} \end{cases} \tag{7}$$

where $\mathbf{S}$ is the set of filtered time slices. Figure 4 shows an example of a sequence in which one observation, y_3, has been filtered out. It is important to note that in such a situation, our algorithm still evaluates the probability of z_3 taking each possible state, and is also still constrained by the aggregate power demand x_3. This ensures the approach is robust even in situations in which the modelled appliance's 'turn on' or 'turn off' observation has been filtered out.

Second, we evaluate the joint probability of all sequences in our model $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ using the product of Equations 1, 2, 4 and 5:

$$p(\mathbf{x}, \mathbf{y}, \mathbf{z}|\boldsymbol{\theta}) = p(z_1|\boldsymbol{\pi}) \prod_{t=2}^{T} p(z_t|z_{t-1}, \mathbf{A}) \prod_{t=1}^{T} P(w_{z_t} \leq x_t|z_t, \boldsymbol{\phi}) \prod_{t \in S} p(y_t|z_t, z_{t-1}, \boldsymbol{\phi}) \tag{8}$$

This is similar to the Viterbi algorithm's joint probability evaluation in a HMM with two exceptions. First, the product over emissions are filtered according to the criteria specified by Equation 7, instead of over full sequence, $1, \ldots, T$. Second, the joint probability over two observation sequences, $\mathbf{x}$ and $\mathbf{y}$ are evaluated, as opposed to just a single sequence in a standard HMM. This allows changes in the aggregate power to determine any likely change of appliance states, while imposing the constraint that appliances are only likely to be 'on' if the observed aggregate power reading is above that appliance's mean power demand.

4 Accuracy Evaluation Using REDD

The proposed approach has been evaluated using the Reference Energy Disaggregation Dataset (REDD) (http://redd.csail.mit.edu/) described by Kolter and Johnson (2011). This data set was chosen as it is an open data set collected specifically for evaluating NIALM methods. The dataset comprises six houses, for which both household aggregate and circuit-level power demand data are collected. Both aggregate and circuit-level data were down sampled to one measurement per minute. We chose to focus on high energy consuming appliance types, for which a single generalisable prior model could be built by a domain expert.

To date, only two other approaches have been benchmarked on this data set. Kolter and Johnson (2011) proposed a supervised approach which requires sub-metered data from all appliances in the house for training and Kolter and Jaakkola (2012) proposed an unsupervised approach which clusters together features extracted from data sampled thousands of times faster than the the data we assume. Our approach does not assume that either sub-metered training data or high frequency sampled aggregate data are available, and therefore a direct performance comparison is not possible. However, we benchmark our training method against two variations of our own approach that demonstrate how our approach is able to use a single prior to generalise across multiple appliance instances. The three approaches used were: a variant of this approach where the prior was not tuned at all (NT), the approach described in this paper where the prior was tuned using only aggregate data (AT), and a variant of this approach where the prior was tuned using sub-metered data (ST).

We use two metrics to evaluate the performance of each approach, one for each of the objectives of NIALM. First, when the objective is to disaggregate the total energy consumed by each appliance over a period of time, we average the normalised error in the total energy assigned to an appliance over all days, as defined by:

$$\left| \sum_t w_t^{(n)} - \sum_t \mu_{z_t^{(n)}}^{(n)} \right| \Big/ \sum_t w_t^{(n)} \tag{9}$$

Second, when the objective is to disaggregate the power demand of each appliance in each time slice, we use the root mean square error, as defined by:

$$\sqrt{\frac{1}{T} \sum_t \left(w_t^{(n)} - \mu_{z_t^{(n)}}^{(n)} \right)^2} \tag{10}$$

Table 1 shows the mean normalised error in the total assigned energy to each appliance per day over multiple houses. It can be seen that the disaggregation error for models trained using aggregate data (AT) as proposed in this paper is comparable to that for models trained using sub-metered data (ST). This result demonstrates the success of the training method by which our approach extracts appliance signatures from the aggregate load. It is interesting to note that the prior models themselves (NT), when not tuned for each household, are not specific enough to disaggregate the appliance's load from other appliance loads. Consequently, the model can match the signatures generated by other appliances and can therefore greatly over-estimate an appliance's total energy consumption. Another point to note is that for the sub-metered variant (ST), it was necessary to add Gaussian noise to the sub-metered data to prevent over fitting, and ensure the model is general enough to match noisier signatures in the aggregate load. However, as a result it can perform worse than the model trained using aggregate data (AT).

Table 2 shows the root mean square error in the power assigned to each appliance in each time slice over multiple houses. It can be seen that similar trends are present in the error in the power in each time slice as the error in the total energy. This confirms that errors which cancel out due to over-estimates and under-estimates in different time slices have not resulted in unrepresentatively accurate estimates of the total energy consumption figures shown in Table 1.

An additional trend shown in both Tables 1 and 2 is that the disaggregation error increases as the number of appliance uses decreases. This is due to the fact that, when extracted from the aggregate signal, few appliance signatures are clean enough to accurately train the prior model to the specific appliance instance. This lack of extracted data can result in trained models which are not general enough to disaggregate the range of behaviour that an appliance might present.

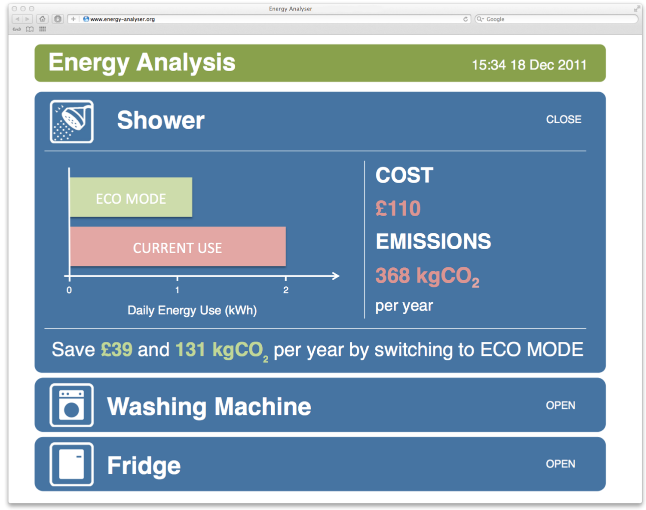


Figure 5: Prototype interface of live deployment

5 Live Deployment of Approach

To demonstrate that our proposed NIALM method is applicable in real scenarios, we collect data in the form of aggregate power measurements logged at one minute intervals from 6 UK households that are fitted with standard UK smart meters. Data is relayed to a central data server through a GPRS modem in each home. However, in a large scale deployment the NIALM system could be embedded in an in-home energy display to avoid issues of data privacy and security. We use a Python wrapper around the core MATLAB disaggregation module to allow the module to be called externally. Our central data server provides the disaggregation module with aggregate power data, and stores the returned disaggregated appliance power data. This information is then presented to the household occupants allowing them to view the energy consumption of many of their appliances.

Figure 5 shows a prototype of the user interface to the system. Using the output of the disaggregation module, the system is able to provide the household occupants with personalised energy saving suggestions. The figure shows a comparison of the energy consumption of the shower in a particular home. To calculate these figures, a prior model is first estimated from the shower's operation manual. This prior model is then trained using the approach presented in this paper and used to disaggregate its energy consumption. Since the shower was used entirely on the 'high' setting, the system could use the prior model to estimate the corresponding energy consumption had the 'eco' setting been used. The potential savings are presented as either energy, financial cost or carbon emission equivalent.

6 Conclusion

In this paper, we have proposed a novel algorithm for training a NIALM system, in which generic models of appliance types can be tuned to specific appliance instances using only aggregate data. We have shown that when combined with a suitable inference mechanism, the models can disaggregate the energy consumption of individual appliances from a household's aggregate load. Through evaluation using real data from multiple households, we have shown that it is possible to generalise between similar appliances in different households. We evaluated the accuracy of our approach using the REDD data set, and have shown that the disaggregation performance when using our training approach is comparable to when sub-metered training data is used. We also presented a deployment of our NIALM system as a real-time application and demonstrated the potential for personalised energy saving feedback. Future work will look at extending the graphical model to include additional information such as time of day and correlation between appliance use. In such a model, the same process of prior training as described in this paper can be applied.

Acknowledgments

We thank the anonymous reviewers for their comments which lead to improvements in our paper. This work was carried out as part of the ORCHID project (EPSRC reference EP/I011587/1) and the Intelligent Agents for Home Energy Management project (EP/I000143/1).

References

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning*. Springer.

Darby, S. 2006. The Effectiveness of Feedback on Energy Consumption, A review for DEFRA of the literature on metering, billing and direct displays. Technical report, University of Oxford, UK.

Department of Energy & Climate Change. 2009. A consultation on smart metering for electricity and gas. Technical report, UK.

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

Kim, H.; Marwah, M.; Arlitt, M. F.; Lyon, G.; and Han, J. 2011. Unsupervised Disaggregation of Low Frequency Power Measurements. In *Proceedings of the 11th SIAM International Conference on Data Mining*, 747–758.

Kolter, J. Z., and Jaakkola, T. 2012. Approximate Inference in Additive Factorial HMMs with Application to Energy Disaggregation. In *International Conference on Artificial Intelligence and Statistics*, 1472–1482.

Kolter, J. Z., and 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*.

Kolter, J. Z.; Batra, S.; and Ng, A. Y. 2010. Energy Disaggregation via Discriminative Sparse Coding. In *Proceedings of the 24th Annual Conference on Neural Information Processing Systems*, 1153–1161.

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

Zeifman, M., and Roth, K. 2011. Nonintrusive appliance load monitoring: Review and outlook. *IEEE Transactions on Consumer Electronics* 57(1):76–84.