Using Hidden Markov Models for Iterative Non-intrusive Appliance Monitoring

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 breaking down a household's total electricity consumption into its contributing appliances. In this paper we propose an approach by which individual appliances are iteratively separated from the aggregate load. Our approach does not require training data to be collected by sub-metering individual appliances. Instead, 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 used to estimate each appliance's load, which is subsequently subtracted from the aggregate load. We evaluate our approach using the REDD data set, and show that it can disaggregate 35% of a typical household's total energy consumption to an accuracy of 83% by only disaggregating three of its highest energy consuming appliances.

1 Introduction

Non-intrusive appliance load monitoring (NIALM), or energy disaggregation, aims to break down a household's aggregate electricity consumption into individual appliances [1]. 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 [2]. Second, should a recommender system seek to inform an occupant of potential savings through deferring appliance use to a time of day when electricity is either cheaper or has a lower carbon footprint, it is necessary to additionally determine the time of use of each appliance. For an energy monitoring system to be both practical and widely applicable, it is essential to take advantage of existing infrastructure rather than designing new hardware. Smart meters constitute an ideal data collection platform for NIALM solutions, and are currently being deployed on national scales.

Recent approaches using machine learning techniques for energy disaggregation from smart meter data fall into two categories. The first group of approaches typically assume that sub-metered (ground truth) data is available to train appliance models prior to performing disaggregation [3]. The second group uses unsupervised disaggregation methods [4, 5], which typically require the labelling of detected appliances, assume knowledge of the number of household appliances or require all the appliances in the home to modelled. Although attractive from a machine learning perspective, such assumptions are impractical in practice and limit many potential applications. It is therefore necessary to design approaches which are capable of disaggregating individual appliances from an aggregate load without explicit knowledge about the type and number of appliance within the home, or access to sub-metered training data.

It is exactly these challenges that we address in this paper, and to do so, we propose a principled approach in which generic prior models of appliance types are tuned to specific appliance instances using only aggregate data from the home in which disaggregation is being performed. We focus on 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. clothes dryer). In our approach,
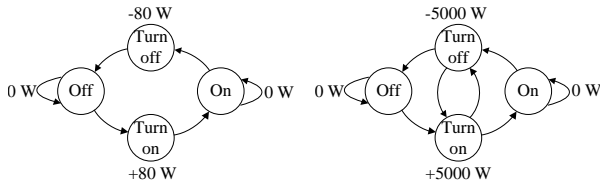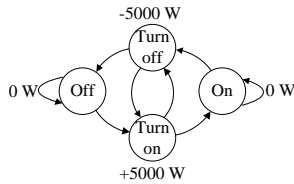
Figure 1: Refrigerator state transition model

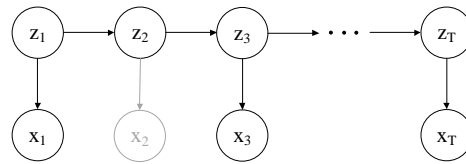Figure 2: Clothes dryer state transition model

Figure 3: HMM where observation $\mathbf{x}_2$ shown in grey has been filtered out

clean appliance signatures are identified within the aggregate signal by applying the Expectation-Maximisation (EM) algorithm to small overlapping windows of aggregate data, which are then used to tune generic models of appliance types to the household's specific appliances. Next, the appliances' loads are modelled using hidden Markov models (HMMs) and disaggregated using an extension of the Viterbi algorithm, before being subtracted from the aggregate load. The extension of the Viterbi algorithm that we use filters the aggregate signal, in parallel to performing disaggregation, such that all other appliances' observations are ignored. These steps are repeated until all appliances for which general models are available have been disaggregated from the aggregate load.

2 Problem description

Formally, the aim of NIALM is as follows. Given a discrete sequence of observed aggregate power readings $\mathbf{x} = \mathbf{x}_1, \ldots, \mathbf{x}_T$, determine the sequence of operational states of each appliance $\mathbf{z}^{(n)} = \mathbf{z}_1^{(n)}, \ldots, \mathbf{z}_T^{(n)}$ where each appliance $n \in \{1, \ldots, N\}$. The operational state of an appliance corresponds to a certain behaviour (e.g. 'on' or 'turning off') and each time slice t represents an instant in time a minute after $t-1$.

3 Energy disaggregation using iterative hidden Markov models

Our approach models each appliance as a hidden Markov model (HMM). In the HMM, each latent discrete variable in the Markov chain represents the state of the appliance at an instant in time. In each time slice, each appliance emits an observation based on its state. A HMM is completely described by three parameters: the probability of the appliance's starting state, $\boldsymbol{\pi}$, the probability matrix of transitions between states, $\mathbf{A}$, and the set of emission functions, ϕ, which in our case are assumed to be Gaussian distributed. These parameters are learnt from aggregate data as described in Section 3.1 and used to disaggregate appliance loads in Section 3.2.

The appliance models described thus far are similar to that used by Kim et al. [5]. However, a key difference in our model is the use of the change in aggregate power demand between consecutive readings as the observation sequence. This results in states which are able to capture both constant and instantaneous changes in each appliance's power demand. Figure 1 shows an 'on-off' state transition diagram used by our approach to model the refrigerator, in which each state represents constant or a change in the appliance's power demand. Figure 2 shows a more complex state transition diagram used to model the clothes dryer. The later model is less restrictive, as it allows 'turn on' states to immediately follow 'turn off' states, which is common behaviour exhibited by the clothes dryer but never the refrigerator, given the sampling rate of aggregate observations. Such modelling has the benefit that Markov states only emit non-zero observations when the appliance's power demand changes, e.g. they turn on or off. This minimises the effect each appliance has on the aggregate observation sequence, and therefore greatly reduces the likelihood that two or more non-zero observations generated by different appliances will occur within the same time slice.

3.1 Training using aggregate data

Our novel training approach takes generic models of appliance types (e.g. all clothes dryers) and trains them to specific appliance instances (e.g. a particular clothes dryer model) using only a household's aggregate power demand. This approach differs from the unsupervised approach used by Kim et al. [5], in that we make use of a prior belief of appliance behaviour and power demand. This training process directly corresponds to learning values for the HMM parameters $\boldsymbol{\pi}$, $\mathbf{A}$ and ϕ.

The generic model of an appliance type consists of prior knowledge of the appliance's state transition matrix and its power demand. The prior state transition matrix consists of a binary matrix, in which ones represent possible transitions between two states. We represent our prior belief of each appliance's demand as expected values of the Gaussian emission function's mean and variance.

Our approach to train such general models to specific appliance instances is as follows. First, data to train the appliance model must be extracted from the aggregate load. This is achieved by running the EM algorithm on small overlapping windows of aggregate data. 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 [6]. As such, the model will reject windows which contain observations which cannot be explained by the appliance model. Therefore, this process effectively identifies windows of aggregated data which were generated by only the modelled appliance. Next, the extracted appliance signatures are used to train the appliance model using a single application of the EM algorithm.

3.2   Disaggregation via extended Viterbi algorithm

The disaggregation task aims to infer each appliance's load given only the aggregate load and the trained appliance's model. We propose an extension of the Viterbi algorithm [7] which is able to filter out 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 in our scenario, it must be robust to other unmodeled appliances contributing emissions to the observed load. To do this, we extend this algorithm by allowing the forward pass to filter out observations for which the joint probability is below a predefined threshold. We then evaluate the joint probability of a sequence by:

$$p(\mathbf{x}, \mathbf{z}|\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{\phi}) = p(\mathbf{z}_1|\boldsymbol{\pi}) \prod_{t=2}^{T} p(\mathbf{z}_t|\mathbf{z}_{t-1}, \boldsymbol{A}) \prod_{t \in S} p(\mathbf{x}_t|\mathbf{z}_t, \boldsymbol{\phi}) \tag{1}$$

This is similar to the Viterbi algorithm's joint probability evaluation apart from the product over emissions, which are only collected over the filtered set of observations, S, instead of the full sequence, $t_1, \ldots, T$. Figure 3 shows an example of a sequence in which one observation, $\mathbf{x}_2$, has been filtered out. It is important to note that in such a situation, the Viterbi algorithm still evaluates the probability of $\mathbf{z}_2$ taking any possible value. 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.

4   Results

The proposed approach has been evaluated using the REDD data set [3]. The dataset comprises six houses, for which both household aggregate and circuit-level power demand data are collected. Both sets of power measurements were down sampled to one measurement per minute to mimic the maximum reporting rate of meters being deployed in the UK smart meter roll out. We chose to disaggregate three of the highest energy consuming appliances: the refrigerator, the microwave and the clothes dryer. Of the six houses, only houses 1, 2 and 3 contained each of these appliances on isolated circuits and had been metered continuously for at least 7 days.

To date, only one other approach [3] has been benchmarked on this data set. Kolter and Johnson's method makes use of sub-metered data to train the model and also requires that all appliances must be modelled. Our approach does not require all appliances to be modelled, and therefore a direct performance comparison is not possible. However, we benchmark against two variations of our own algorithm that mimic the unsupervised and sub-metered training approaches.

We discuss each of the two objectives of NIALM separately. First, when the aim is to inform the household occupants which appliances are consuming the most energy, performance should be assessed according to how accurately a NIALM can infer the actual energy consumption of each appliance. Figure 4 shows a comparison for illustrative purposes of the estimated and actual disaggregated energy as would be provided to the household occupants of house 1. It is interesting to note that just 3 appliances account for 35% of the household's total energy consumption, of which 83% was correctly disaggregated by our approach. Table 1 shows how accurately our approach estimated the energy consumption of each individual appliance, averaged over the three houses. It can be seen that the disaggregation error for models trained using aggregate data is comparable to
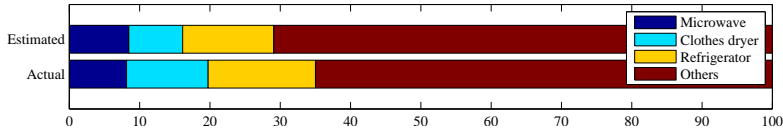
Figure 4: Estimated and actual percentage of disaggregated energy of house 1

Table 1: Disaggregation error averaged over houses 1-3

Appliance	No training	Aggregate training	Sub-metered training
Refrigerator	55%	15%	14%
Clothes dryer	428%	28%	24%
Microwave	54%	22%	10%

Table 2: Usage classification rates averaged over houses 1-3 (Corr. class. = Percentage actual uses correctly classified, False det. = Percentage detected uses which were false)

Appliance	No training		Aggregate training		Sub-metered training	
	Corr. class.	False det.	Corr. class.	False det.	Corr. class.	False det.
Refrigerator	48%	23%	74%	24%	81%	15%
Clothes dryer	38%	77%	89%	6%	94%	6%
Microwave	24%	36%	71%	24%	78%	20%

that for models trained using sub-metered data. Table 1 also shows the improvement in performance after training using aggregate data compared to just using the generic models.

Second, when the aim is to identify the uses of each appliance, a different evaluation metric is required. Unlike the previous application, rare misclassifications do not cancel out over time, and instead might result in incorrect advice being given to the household occupants. Table 2 shows the percentage of actual appliances uses that were correctly classified and the percentage of detected uses that were false, for each training method averaged across the three houses. Again, the performance of approach trained using aggregate data is comparable to that using sub-metered data.

5   Conclusion

In this paper, we have proposed an approach to NIALM capable of independently disaggregating the energy consumption of individual appliances from a household's aggregate load. We have proposed a novel method by which generic models of appliance types can be tuned to specific appliance instances using only aggregate data. Through evaluation using real aggregate and circuit-level data from three households, we have shown that similar performance can be achieved by tuning the generic models using aggregated data compared to using sub-metered data. Future work will look at extending this approach to more high energy consuming appliances using data from a greater number of households. We also aim to investigate the gain of disaggregating appliances in parallel, instead of iteratively, to resolve conflicts caused by two or more appliances changing state simultaneously.

References

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

[2] S. Darby. 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, 2006.

[3] J. Z. Kolter and M. J. Johnson. REDD: A Public Data Set for Energy Disaggregation Research. In *Workshop on Data Mining Applications in Sustainability (SIGKDD)*, San Diego, CA, 2011.

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

[5] H. Kim, M. Marwah, M. F. Arlitt, G. Lyon, and J. Han. Unsupervised Disaggregation of Low Frequency Power Measurements. In *11th International Conference on Data Mining*, pages 747–758, Arizona, 2011.

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

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