A Hierarchical Hidden Markov Model Framework for Home Appliance Modeling

Weicong Kong, *Student Member, IEEE*, Zhao Yang Dong, *Senior Member, IEEE*,
David J. Hill, *Life Fellow, IEEE*, J. Ma, J. H. Zhao, and F. J. Luo

***Abstract*—Correctly anticipating load characteristics of low voltage level is getting increased interest by distribution network operators. Energy disaggregation could be one of the potential approaches to exploit the massive amount of smart meter data to fulfill the task. Proper individual home appliance modeling is critical to the performance of NILM. In this paper, a hierarchical hidden Markov model (HHMM) framework to model home appliances is proposed. This model aims to provide better representation for those appliances that have multiple built-in modes with distinct power consumption profiles, such as washing machines and dishwashers. The dynamic Bayesian network representation of such an appliance model is built. A forward–backward algorithm, which is based on the framework of expectation maximization, is formalized for the HHMM fitting process. Tests on publically available data show that the HHMM and proposed algorithm can effectively handle the modeling of appliances with multiple functional modes, as well as better representing a general type of appliances. A disaggregation test also demonstrates that the fitted HHMM can be easily applied to a general inference solver to outperform conventional hidden Markov model in the estimation of energy disaggregation.**

***Index Terms*—Hierarchical hidden Markov model, energy disaggregation, residential load modelling, expectation maximization.**

I. Introduction

RESIDENTIAL demand account for about from 22% to 50% of the total energy consumption [1], [2]. However, consumer behavior at the residential level are relatively unknown in traditional system planning and design. The aggregated load profile may attract more attention in transmission level, but the development of increasing distributed renewable generation, smart grid applications and energy storage systems have drawn increasing interests on the behaviour of previously neglected end users. The advent of smart meter infrastructure has provided plentiful information for studying load characteristics, but research in mining such big data to provide actionable insights is still at the early stage.

Load disaggregation is one of the key directions for effective use of existing smart meter infrastructure, because it has great potential to link to many useful applications such as energy awareness and energy conservation, controllable load quantitative evaluation, human behaviour and load prediction. This idea dated from Hart in 1992 [3]. Building on that, many researchers had contributed on high-frequency electric characteristics such as current and voltage waveforms harmonics and start-up transients [4]–[9].

Given the infrastructure, however, the high-frequency features are not easily accessible with existing smart meters, which limit their practicality. For example, according to Australian smart metering infrastructure [10], the only readily available data from smart meters are real power consumption at 5-second intervals. In this case, models that utilizing such low-frequency sampling data are attracting increasingly more attention. The hidden Markov model (HMM) has been the most common model being used to handle low-frequency real power consumption data [11]–[13]. Lots of variant HMM based models were presented to model home appliances such as conditional HMM [11], hidden semi-Markov model (HSMM) [11], conditional HSMM [11], difference HMM [14], and the difference HSMM [15].

Despite so many HMM based models being proposed to target the energy disaggregation problem, all the above mentioned models have difficulty to distinguish the standby states (those states that only consume relatively little power during active operation) and the OFF states. For example, during the operations of a dishwasher or a washing machine, there are certain periods when water is being filled and the appliance is idle and waiting for the next process. These periods should be distinguished from the OFF states. Moreover, to the best of our knowledge, models that directly face the modelling challenge for appliances with multiple operation modes have not yet been discussed before. Commonly, a single HMM is usually built for one appliance from a power sequence that covers

Manuscript received April 8, 2016; revised August 23, 2016 and October 27, 2016; accepted November 3, 2016. Date of publication November 8, 2016; date of current version June 19, 2018. This work was supported in part by the Department of Education Australia through Endeavour Postgraduate Scholarship scheme, in part by the Sydney University Bridge Grant and the Faculty Research Cluster Program, and in part by the China Southern Power Grid Company under Project WYKJ00000027. The work of F. J. Luo was supported by the Early Career Research Development Scheme of Faculty of Engineering and Information Technology, University of Sydney, Australia. Paper no. TSG-00449-2016.

W. C. Kong and J. Ma are with the School of Electrical and Information Engineering, University of Sydney, Sydney, NSW, Australia (e-mail: weicong.kong@sydney.edu.au; j.ma@sydney.edu.au).

Z. Y. Dong is with the School of Electrical and Information Engineering, University of Sydney, Sydney, NSW, Australia, and also with China Southern Grid Company, Guangzhou, China (e-mail: zydong@ieee.org).

D. J. Hill is with the School of Electrical and Information Engineering, University of Sydney, Sydney, NSW, Australia, and also with the University of Hong Kong, Hong Kong (e-mail: david.hill@sydney.edu.au).

J. H. Zhao is with the Chinese University of Hong Kong Shenzhen, Shenzhen, China (e-mail: junhua.zhao@outlook.com).

F. J. Luo is with the School of Civil Engineering, University of Sydney, Sydney, NSW, Australia (e-mail: fengji.luo@sydney.edu.au).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSG.2016.2626389

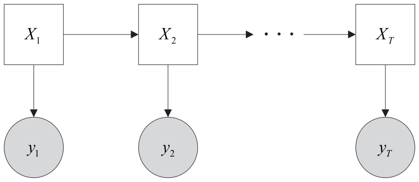


Fig. 1. Graphical illustration of a HMM.

its consumption for days or even weeks. For multi-mode appliances, distinct power profiles from the different modes may or may not be captured, depending on users' behaviour. A HMM will then be fit with this long concatenated power sequence. In this case, the re-usability of such models is compromised in the load disaggregation problem, because it cannot correctly represent the power consumption even for the specific appliances with known brands and makes in a previously unseen household.

In this paper, a HMM variant with hierarchical structure is proposed to enhance the representation power of Markov chain based models, especially for multi-mode appliances. The power profiles of an appliance with multiple modes is just like the pronunciation signals of a word being pronounced with multiple accents. The hierarchical hidden Markov model (HHMM) was introduced in the area of speech recognition to address the problem of multiple accents of a single word [16]. Inspired by this, we propose to adopt the HHMM structure for appliance modelling to encode the features of multi-mode appliances and differentiate the operating standby states from the OFF states. In this case, the proposed model consists of two Markov chains, namely the upper chain which depicts the *mode* of the appliances and the lower chain which models the power consumption profile of the specific mode.

The rest of the paper is organized as follows. Section II reviews both the HMM and HHMM. Section III is the main contribution of this paper, where it extends the expectation maximization model fitting framework from standard HMM to the proposed HHMM model. In Section IV, the proposed model and the proposed fitting algorithm are applied to show that our proposed model can work with measurement granularity as low as 1 sample per minute and outperform conventional HMM in a bootstrap filtering load disaggregation solver. Section V concludes this paper.

II. Model Representations

A. Hidden Markov Model (HMM)

HMM provides a good framework to describe the dynamic through discretised time series of individual appliances. The graphical illustration of HMM is given as Fig. 1. Under this framework, the real power consumption observations or measurements are represented as $y_{1:T} = \{y_1, \dots, y_T\}$, each of which only depends on some unobserved or latent state x_t of the appliance at time slice t. The variable T is the total length of the aggregate power sequence, which can be naturally defined by the updating frequency of the smart meter infrastructure. For example, if the smart meters tend to push the local readings to utilities every 30 minutes, then $T = 30$ in the case of minute-to-minute sampling interval. The Markovianity is defined such that state X_{t+1} is only dependent on state X_t. Normally, home appliances can be reasonably represented as finite state machines, such as ON-OFF machines or three-state machines, so the state variables in the graphical representation are discretised and represented with un-shaded (unobserved) squares (discretised). The observations of real power are continuous variables emitted by the unobserved states, which are represented with shaded (observed) circles (continuous).

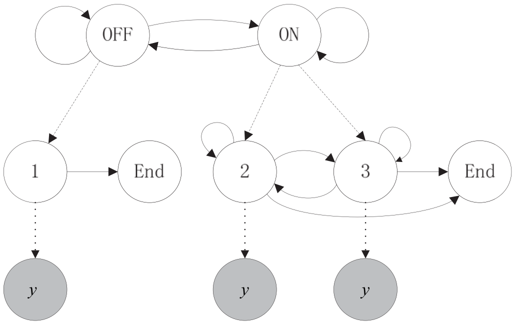


Fig. 2. The state transition diagram for a two-level HHMM, modelling a 3-state appliance. (Solid arrows represent horizontal transitions between states; dashed arrows represent vertical transitions; dotted arrows triggers emissions.)

The parameterization of a HMM is given by three parameters:

$$\pi_i = P(X_1 = i) \tag{1}$$

$$A_{i,j} = P\big(X_t = j \big| X_{t-1} = i\big) \tag{2}$$

$$P\big(y_t \big| X_t = i\big) \sim \mathcal{N}(\mu_i,\ \sigma_i) \tag{3}$$

where $\{\pi_i : i = 1, \dots, K\} = \boldsymbol{\pi} \in \mathbb{R}^K$ is categorical distribution for the initial state, $A_{i,j}$ is the corresponding element of the state transition matrix $\boldsymbol{A}$, and y_t means the Gaussian emission distribution for the ith hidden state, we use $\boldsymbol{O}_i = \{\mu_i, \sigma_i\}$ to denote the set of observation parameters when the hidden variable is at the ith state.

B. Hierarchical Hidden Markov Model (HHMM)

Fine *et al.* [16] introduced an extension of the HMM to model problems with hierarchical structure of multiple levels. This model was further developed to have a dynamic Bayesian network form in [17]. Inspired by the generative structures, it can be more accurate for an electric appliance to be modelled as a two-level HHMM. The state transition diagram of an HHMM for an arbitrary 3-state appliance is shown as Fig. 2. The upper level models the ON-OFF state of an appliance, while the lower level models the operating states and their observation emissions. The fundamental idea of HHMM is that the states at the upper level emit sequences rather than observations of power. Such states are called "abstract" states [17]. On the other hand, the states at the lower level can produce single observations, which are called "concrete" states [17]. The observations are governed by each of the sub-HMMs. Moreover, there is an end state for each sub-HMM. The generative process of a sequence of power observations for such a model is illustrated with Fig. 2. We assume that the abstract state is ON in the beginning. The ON abstract state calls its

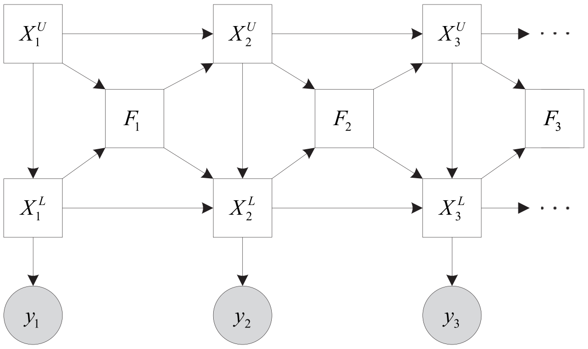


Fig. 3. Graphical illustration of a two-level HHMM.

sub-HMM, entering into either state 2 or 3 depending on the probability parameters. This entering process from the upper level is called a "vertical" transition. Suppose it enters state 2. Since state 2 is a concrete state, it emits an observation y at the current time step. At the next time step, the state may remain at state 2, or transit to either state 3 or the end state for this sub-HMM. All these transitions are called "horizontal" transitions. After a number of time steps, the state may enter the end state at the lower level. Once this happens, it returns the control to the parent state which called this sub-HMM. In this case, it returns to the abstract ON state, then the horizontal transition process continues on the upper level.

According to the generative process, the dynamic Bayesian network representation of such a two-level HHMM can be drawn as Fig. 3.

In Fig 3, X_t^U represents the state variable at the upper level at time t, and X_t^L is the state variable at the lower level. F_t is a binary auxiliary variable which has the value of 1 if the lower level HMM has just finished at time t, otherwise it is 0. The states at the lower level emit a single power observation at each time step. The conditional probability distributions (CPDs) of each node types are defined as follows.

Firstly, at the lower level, X_t^L is governed by the parameters of the sub-HMM that X_t^U calls. The conditional probability distribution of X_t^L can be written as follows:

$$P\left(X_t^L = j \middle| X_{t-1}^L = i\,, F_{t-1} = f, X_t^U = \kappa\right) = \begin{cases} \tilde{A}_\kappa^L(i,j) & \text{if } f = 0 \\ \pi_\kappa^L(j) & \text{if } f = 1 \end{cases} \quad (4)$$

where it is assumed that $i, j \neq end$, where end represents the end state for this HMM at the lower level; κ is the state that the upper level is at; π_κ^L is the distribution for initial state for the lower level given that the parent variable is in state κ. Similarly, A_κ^L is the state transition matrix for the sub-HMM given that the parent variable is in state κ. Unlike the traditional HMM, the HHMM transition matrix should include the transition probability from an arbitrary state to the end state. Therefore, $\tilde{A}_\kappa^L$, which is a rescaled version of A_κ^L, is introduced in the equation. The relationship between the sub-HMM transition matrix and the rescaled matrix satisfies the following equation:

$$\tilde{A}_\kappa^L(i,j)\left(1 - A_\kappa^L(i, end)\right) = A_\kappa^L(i,j) \quad (5)$$

where $A_\kappa^L(i, end)$ is the probability of a state being terminated from state i for the sub-HMM. With (5), the scaled transition matrix becomes stochastic, i.e., $\sum_{j=1}^K \tilde{A}_\kappa^L(i,j) = 1$, where K is the number of states for the sub-HMM.

Since the sub-HMM is required to enter the end state before F_t being set to 1, we have

$$P\left(F_t = 1 \middle| X_t^L = i, X_t^U = \kappa\right) = A_\kappa^L(i, end) \quad (6)$$

$$P\left(F_t = 0 \middle| X_t^L = i, X_t^U = \kappa\right) = 1 - A_\kappa^L(i, end) \quad (7)$$

The conditional probability of the observation y_t is also defined using a Gaussian distribution. The parameter set $\boldsymbol{O}_i$ denotes the parameters for $P(y_t|X_t^L = i)$.

On the other hand, the transitions on the upper level are governed by one transition matrix, but they can only change state when the F_{t-1} signals the lower-level HMM has finished. Otherwise, the upper level should remain in the same state as in the previous time step. This can be written as follows:

$$P\left(X_t^U = \kappa \middle| X_{t-1}^U = \eta, F_{t-1} = f\right) = \begin{cases} \delta(\eta,\kappa) & \text{if } f = 0 \\ A^U(\eta,\kappa) & \text{if } f = 1 \end{cases} \quad (8)$$

where $\delta(i,j)$ is the Kronecker delta function which has the value one if the two parameters are identical; A^U is the transition matrix for the upper-level HMM.

The initial state of the upper level is defined:

$$P\left(X_1^U = \kappa\ \right) = \pi^U(\kappa) \quad (9)$$

where $\pi^U(\kappa)$ is prior distribution for the upper-level HMM.

III. Model Fitting

The model fitting process is basically to estimate a set of model parameters which yields the highest observation likelihood. Let $\boldsymbol{\theta}$ denote the set of parameters. In the case of a HMM, $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \mathbf{A}, \mathbf{O}\}$ according to the definition of HMM in the previous section. Let $\boldsymbol{y} = y_{1:T}$ denote the whole sequence of observation, the model fitting process can be represented as

$$\overline{\boldsymbol{\theta}} = \text{argmax}_{\boldsymbol{\theta}} P(\boldsymbol{y}|\boldsymbol{\theta}) \quad (10)$$

where $\overline{\boldsymbol{\theta}}$ is the estimated set of parameters for the model.

The most common approach to estimate $\overline{\boldsymbol{\theta}}$ is through the well famous expectation maximization (EM) algorithm [18]. In the following sections, we will first review the model fitting process for HMM using EM, and then further generalize to fitting HHMM via EM. Our HHMM fitting framework is more concise and consistent with the HMM fitting framework by adopting the dynamic Bayesian network representation, where only two variables are needed instead of four in the fitting framework of [16].

Note that all the variables in the following sections are derived with parameter set $\boldsymbol{\theta}$. We remove $\boldsymbol{\theta}$ from the probability notation for simplicity such that $P(\boldsymbol{y}|\boldsymbol{\theta}) = P(y_{1:T})$.

A. Expectation Maximization for HMM

1) Forward-Backward Variables: Firstly, the forward variable is defined as:

$$\alpha_t(j) = P(X_t = j, y_{1:t}) \quad (11)$$

Alternatively, the forward variable is usually denoted in the normalized version such that:

$$\begin{aligned}\alpha_t(j) &\triangleq \frac{P(X_t = j, y_{1:t})}{P(y_{1:t})} = P\big(X_t = j \big| y_{1:t}\big)\\ &\propto \left(\sum_{i=1}^{K} \alpha_{t-1}(i) A_{i,j}\right) P\big(y_t \big| X_t = j\big) \qquad (12)\end{aligned}$$

where $P(y_t|x_t = j)$ is the probability distribution at t when the model is at state j. In this paper, we adopt the second version of forward variable.

The backward variable can be calculated recursively as:

$$\beta_t(i) \triangleq P\big(y_{t+1:T} \big| X_t = i\big) = \sum_{j=1}^{K} A_{i,j} P\big(y_{t+1} \big| X_{t+1} = j\big) \beta_{t+1}(j) \qquad (13)$$

Additionally, the base case for both forward and backward variables is defined:

$$\alpha_1(j) = P\big(y_1 \big| X_1 = j\big) \pi_j \qquad (14)$$

$$\beta_T(i) = 1 \qquad (15)$$

2) Expectation Maximization: Once the forward and backward variables are defined, the expectation maximization (EM) algorithm can be applied. Firstly, the expectation calculation step is outlined. The probability of each state at each time step given the whole observation sequence can then be calculated:

$$\begin{aligned}\gamma_t(i) &\triangleq P\big(X_t = i \big| y_{1:T}\big)\\ &= \frac{1}{P(y_{1:T})} P\big(y_{t+1:T} \big| X_t = i\big) P\big(X_t = i \big| y_{1:t}\big)\\ &\propto \alpha_t(i) \beta_t(i) \qquad (16)\end{aligned}$$

In order to implement the Expectation Maximization (EM) algorithm, the two-slice distribution variable is also needed as follows:

$$\begin{aligned}\xi_t(i,j) &\triangleq P\big(X_{t-1} = i, X_t = j \big| y_{1:T}\big)\\ &\propto P\big(X_t = j \big| X_{t-1} = i\big) \times P\big(X_{t-1} = i \big| y_{1:t-1}\big)\\ &\quad \times P\big(y_{t+1:T} \big| X_t = j\big) \times P\big(y_t \big| X_t = j\big)\\ &= A_{i,j} \alpha_{t-1}(i) \beta_t(j) P\big(y_t \big| X_t = j\big) \qquad (17)\end{aligned}$$

Calculation of γ_t and ξ_t forms the estimation step (E step) of the EM algorithm.

In the maximization step (M step), all parameters of the HMM are updated correspondingly:

$$\overline{\pi}_i = \gamma_1(i) \qquad (18)$$

$$\overline{A}_{i,j} = \frac{1}{Z} \sum_{t=2}^{T} \xi_t(i,j) \qquad (19)$$

where Z is a normalizing constant to ensure the transition matrix is a stochastic matrix.

Parameters μ_i and σ_i are updated by fitting a Gaussian distribution using a maximum likelihood estimate (MLE) to the contribution of the observation sequence. The contribution at time t can be calculated by:

$$\overline{y}_t(i) = \gamma_t(i) \times y_t. \qquad (20)$$

3) Numerical Underflow: Numerical underflow is typical when the joint probability keeps decreasing due to continuous multiplication of later probability as the length of the sequence grows. The value can exceed the maximum length that a floating point number can represent in a computer, resulting in those values to be considered as 0.

To deal with this issue, all the probability multiplications are transformed to summation in log space, so are all the forward and backward variables. Further, normalization is also performed in log space to mitigate the effect, so that $\sum_j \exp(\alpha_t(j)) \ = 1$ at each time step.

4) The Log-Likelihood of Model Parameters: The likelihood of the observation sequence is $P(y_{1:T}|\boldsymbol{\theta})$. Recalling that $\boldsymbol{\theta}$ is removed for simplicity, the observation likelihood is then $P(y_{1:T})$.

In fact, the calculation of $P(y_{1:T})$ can be derived through the definition of the forward variable (12). The forward variables can be are rewritten into the following form

$$\begin{aligned}\alpha_t(j) &= P\big(X_t = j \big| y_{1:t}\big)\\ &= \frac{1}{P(y_t|y_{1\,:\,t-1})} P\big(X_t = j, y_t \big| y_{1\,:\,t-1}\big) \qquad (21)\end{aligned}$$

Letting $c_t = P(y_t|y_{1:t-1})$, it can be also calculated by marginalizing out all possible states.

$$c_t = \sum_j P\Big(X_t = j, y_t \Big|_{1\,:\,t-1}\Big) \qquad (22)$$

Because the following equation holds

$$P\big(X_t = j, y_t \big| y_{1:t-1}\big) = \left(\sum_{i=1}^{K} \alpha_{t-1}(i) A_{i,j}\right) P\big(y_t \big| X_t = j\big)$$

it can be seen that c_t is actually a normalising constant for the forward variables at the time step t. Therefore, the observation likelihood can be calculated by tracking and multiplying all normalizing constants as follows

$$\begin{aligned}P(y_{1:T}) &= P(y_1) P\big(y_2 \big| y_1\big) \cdots P\big(y_T \big| y_{1:T-1}\big)\\ &= \prod_{t=1}^{T} c_t \qquad (23)\end{aligned}$$

Once the new likelihood is updated, a convergence test is made to decide whether to terminate or to repeat the E-step. The complete framework of the EM algorithm is shown as Fig. 4.

B. Expectation Maximization for HHMM

Similarly, the EM approach for HMM can be modified and generalized to learn a HHMM. Due to the additional layer, the formulation of EM algorithm for a HHMM is non-trivial. Let $\boldsymbol{\Theta}$ represent the set of parameters for a HHMM, for all values of κ, we have

$$\boldsymbol{\Theta} = \Big\{A^U, \pi^U, \Big\{\tilde{A}^L_\kappa\Big\}, \big\{\pi^L_\kappa\big\}, \big\{A^L_\kappa(:, end)\big\}, \boldsymbol{O}\Big\} \qquad (24)$$

In our HHMM, it is assumed that each level has its unique set of state labels. For example, on the upper level, the state label could be $\{ON, OFF\}$, while on the lower level, the state label could be $\{1, 2, 3\}$ as shown in Fig. 2. In this case, the

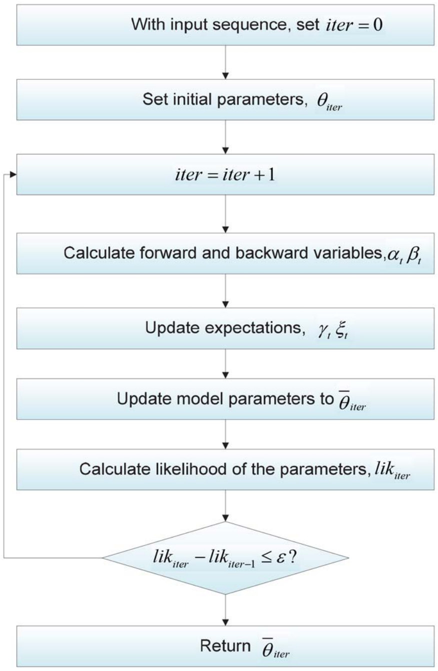

With input sequence, set $iter = 0$
Set initial parameters, θ_{iter}
$iter = iter + 1$
Calculate forward and backward variables, α_t β_t
Update expectations, γ_t ξ_t
Update model parameters to $\bar{\theta}_{iter}$
Calculate likelihood of the parameters, lik_{iter}
$lik_{iter} - lik_{iter-1} \le \varepsilon$?
Return $\bar{\theta}_{iter}$

Fig. 4. The EM algorithm framework.

vertical transition probability matrix could be sparse because the lower level state is definitely 1 when the upper level is OFF, i.e., $P(X_t^L = 1|X_t^U = OFF) = 1$, while $(X_t^L = 2|X_t^U = OFF)$ and $(X_t^L = 3|X_t^U = OFF)$ are definitely 0. According to this state labelling setting, the forward and backward variables for HHMM can be similarly formulated in the following sections.

1) The Forward Variables:

$$\begin{aligned}\alpha_t(j,\kappa,f) &\triangleq P\big(X_t^L = j, X_t^U = \kappa, F_{t-1} = f \big| y_{1:t}\big)\\ &\propto P\big(y_t \big| X_t^L = j\big) \sum_i \sum_\eta h(i,\eta) \sum_g \alpha_{t-1}(i,\eta,g)\end{aligned} \tag{25}$$

where $h(i,\eta)$ is an auxiliary function such that

$$h(i,\eta) = \begin{cases} \Big(1 - A_\eta^L(i,end)\Big)\delta(\eta,\kappa)\tilde{A}_\kappa^L(i,j) & \text{if } f = 0\\ A_\eta^L(i,end)A^U(\eta,\kappa)\pi_\kappa^L(j) & \text{if } f = 1\end{cases} \tag{26}$$

Additionally, the initial forward variable is defined as:

$$\alpha_1(j,\kappa,f) = \begin{cases} 0 & \text{if } f = 0\\ \pi^U(\kappa)\pi_\kappa^L(j)P\big(y_1\big|X_1^L = j\big) & \text{if } f = 1\end{cases} \tag{27}$$

This is because F_0 is assumed to be 1, so that an HHMM will enter its initial state at the upper level according to the upper-level prior distribution.

2) The Backward Variables:

$$\begin{aligned}\beta_t(i,\eta,f) &\triangleq P\big(y_{t+1:T}\big|X_t^L = i, X_t^U = \eta, F_{t-1} = f\big)\\ &= \sum_j P\big(y_{t+1}\big|X_{t+1}^L = j\big) \sum_\kappa \bigg(\Big(1 - A_\eta^L(i,end)\Big)\\ &\quad \times \delta(\eta,\kappa) \times \tilde{A}_\kappa^L(i,j) \times \beta_{t+1}(j,\kappa,0) + A_\eta^L(i,end)\\ &\quad \times A^U(\eta,\kappa) \times \pi_\kappa^L(j) \times \beta_{t+1}(j,\kappa,1)\bigg)\end{aligned} \tag{28}$$

Similarly, the initial backward variable is given as:

$$\beta_T(i,\eta,f) = 1 \tag{29}$$

3) Sufficient Statistics: On the lower level, the following sufficient statistics at each time step given the whole sequence of observations are needed for the E-step. Their calculations are given respectively as follows.

Horizontal transition probability

$$\begin{aligned}\xi_t^h(i,j,\kappa) &= P\Big(X_{t-1}^L = i, X_t^L = j, X_t^U = \kappa, F_{t-1} = 0\big|y_{1:T}\Big)\\ &\propto \sum_\eta \sum_g \alpha_{t-1}(i,\eta,g) \times \Big(1 - A_\eta^L(i,end)\Big)\\ &\quad \times \delta(\eta,\kappa) \times \tilde{A}_\kappa^L(i,j) \times P\big(y_t\big|X_t^L = j\big)\\ &\quad \times \beta_t(j,\kappa,0)\end{aligned} \tag{30}$$

Vertical transition probability

$$\begin{aligned}\xi_t^v(j,\kappa) &= P\big(X_t^L = j, X_t^U = \kappa, F_{t-1} = 1\big|y_{1:T}\big)\\ &\propto \alpha_t(j,\kappa,1)\beta_t(j,\kappa,1)\end{aligned} \tag{31}$$

Smoothed state probability

$$\begin{aligned}\gamma_t(j,\kappa,f) &= P\big(X_t^L = j, X_t^U = \kappa, F_{t-1} = f\big|y_{1:T}\big)\\ &\propto \alpha_t(j,\kappa,f)\beta_t(j,\kappa,f)\end{aligned} \tag{32}$$

End transition probability

$$\begin{aligned}\xi_t^e(i,\eta,1) &= P\big(X_t^L = i, X_t^U = \eta, F_t = 1\big|y_{1:T}\big)\\ &\propto \sum_j \sum_\kappa \sum_f \alpha_t(i,\eta,f) \times A_\eta^L(i,end)\\ &\quad \times A^U(\eta,\kappa) \times \pi_\kappa^L(j)\\ &\quad \times P\big(y_t\big|X_t^L = j\big) \times \beta_{t+1}(j,\kappa,1)\end{aligned} \tag{33}$$

where $\xi_t^e(i,\eta,1)$ denotes the probability that the sub-HMM ends when the lower level is at the ith state and the upper level is at the ηth state. In order to normalize the end transition parameters, the probability that the sub-HMM does not end for the same set of states is also needed, which is

$$\begin{aligned}\xi_t^e(i,\eta,0) &= P\big(X_t^L = i, X_t^U = \eta, F_t = 0\big|y_{1:T}\big)\\ &\propto \sum_j \sum_\kappa \sum_f \alpha_t(i,\eta,f)\\ &\quad \times \Big(1 - A_\eta^L(i,end)\Big) \times \delta(\eta,\kappa)\\ &\quad \times \tilde{A}_\kappa^L(i,j) \times P\big(y_{t+1}\big|X_{t+1}^L = j\big)\\ &\quad \times \beta_{t+1}(j,\kappa,0)\end{aligned} \tag{34}$$

On the other hand, the horizontal transition probability is also needed for the upper level:

$$\begin{aligned}\chi_t(\eta,\kappa) &= P\Big(X_{t-1}^U = \eta, X_t^U = \kappa, F_{t-1} = 1\big|y_{1:T}\Big)\\ &\propto \sum_i \sum_j \sum_g \alpha_{t-1}(i,\eta,g) \times A_\eta^L(i,end)\\ &\quad \times A^U(\eta,\kappa) \times \pi_\kappa^L(j)\\ &\quad \times P\big(y_t\big|X_t^L = j\big) \times \beta_t(j,\kappa,1)\end{aligned} \tag{35}$$

TABLE I
THE PROPOSED EXPECTATION MAXIMIZATION ALGORITHM FOR HHMM

```
Algorithm I: The EM algorithm for HHMM
Inputs: $\boldsymbol{y}$, $K^U$, $K^L = \{K_1^L, \dots, K_\kappa^L\}$
Outputs: $\boldsymbol{\Theta}$
Set $iter = 0$
Initialize $\boldsymbol{\Theta}_{\text{iter}}$
Set $iter = iter + 1$
While $iter < maxIter$ and $\Delta lik < threshold$
    For $t = 1$ to $T$
        // the E-step
        Compute and store forward and backward variables $\alpha_t, \beta_t$ using (25), (27), (28), (29)
        Compute and store all sufficient statistics $\xi_t^h, \xi_t^v, \gamma_t, \xi_t^e, \chi_t$ using (30), (31), (32), (33), (34), (35)
        // the M-step
        Update $\boldsymbol{\Theta}$ using (36), (37), (38), (40), (41)
        Update the model log-likelihood $loglik_{iter}$ using (23)
        Set $iter = iter + 1$
    End for
End while
Return $\boldsymbol{C_1}, \dots, \boldsymbol{C_{N_c}}, \boldsymbol{\mu}, \boldsymbol{\sigma}$
```

4) The Maximization Step: With all sufficient statistics calculated in the E-step, HHMM parameters can then be updated in the subsequent M-step similarly to the HMM M-step. All the parameters are updated according to the following equations respectively.

Firstly, the contribution of an observation to a concrete state i is calculated in a similar way as for HMM. That is

$$\overline{y}_t(i) = \left(\sum_{\kappa} \gamma_t(i, \kappa) \right) \times y_t \tag{36}$$

The lower level transition parameters are updated as:

$$\overline{\overline{A}}_{\kappa}^{L}(i, j) = \frac{1}{Z_h} \sum_{t=2}^{T} \xi_t^h(i, j, \kappa) \tag{37}$$

where Z_h is the normalizing constant to ensure that $\overline{\overline{A}}_{\kappa}^{L}$ is a stochastic matrix. The end transition parameters are

$$\overline{A}_{\kappa}^{L}(i, end) = \frac{1}{Z_e} \sum_{t=1}^{T-1} \xi_t^e(i, \kappa, 1) \tag{38}$$

where Z_e is the normalizing constant matrix for the end transitions and it can be calculated by

$$Z_e(i, \eta) = \sum_{t=1}^{T-1} \sum_{g} \xi_t^e(i, \kappa, g) \tag{39}$$

The initial state distribution parameters at the lower level are:

$$\pi_{\kappa}^{L}(j) = \frac{1}{Z_v} \sum_{t=1}^{T} \xi_t^v(j, \kappa) \tag{40}$$

The transition parameters at the upper level are:

$$A^U(\eta, \kappa) = \frac{1}{Z_U} \sum_{t=2}^{T} \chi_t(\eta, \kappa) \tag{41}$$

The general framework of the EM algorithm for HHMM is similar to the one for basic HMM shown in Fig. 4. The EM algorithm for HHMM is summarized as Algorithm I which is given in Table I.

C. Complexity Analysis

It can be easily derived from the above formulation that, at each iteration, the EM algorithm has time complexity $\mathcal{O}(K^2T)$ and space complexity $\mathcal{O}(KT)$, while training a HHMM using the EM algorithm has time complexity $\mathcal{O}((2K_UK_L)^2T)$ and space complexity $\mathcal{O}(2K_UK_LT)$, where K is the state dimension of conventional HMM, and K_U, K_L are the upper state dimension and the lower state dimension respectively in our proposed model.

IV. MODELLING REAL-WORLD APPLIANCE WITH HHMM

In this section, we will illustrate modelling real-world appliances with our proposed model and fitting algorithm.

In the research area of load disaggregation, it is worth noting that most prior research either did not clearly state the length of the power profiles from individual appliances they use to fit the models [11], [12], [19], or used a rather lengthy sequence to fit the models [13], [20] before applying them in disaggregation tasks. We want to emphasize that lengthy submetered sequences may not represent good generality, because the extra information such as frequency of use, mode preference, appliance dependency and time-of-use pattern could be inconsistent with previously unseen households and are captured and encoded to the model. Appliance models fitted by lengthy observation sequences cannot be directly applied to new households even though the same types and brands of appliances are used. Therefore, we propose to train each of our appliance models with *only one* typical duty cycle, which eliminating the possibility of including the abovementioned extra information that may compromise the model generality. In this case, we can survey a new household for the brands and types of the appliances and re-use the specific appliance models for a new disaggregation task.

We test our proposed model using publically available data introduced by [21]. Unlike other popular residential load datasets, this dataset provides well-illustrated examples of electric appliances which have multiple operating modes. For each mode of every selected major appliance, a typical duty cycle of power consumption profile was given, which is ideal for testing our proposed HHMM. For example, washing machines normally have cycles such as normal wash, permanent press and delicate wash. The power consumption pattern for each of the cycle differs from each other. The examples are illustrated by the blue lines in Fig. 7 - Fig. 9, showing the real power consumption of different cycles of a same washing machine (LG-WM2016CW1) down-sampled to 1 reading per minute.

A. Fitting Single-Mode Appliances and Its Benefit

Firstly, we demonstrate the benefit of modelling a single-mode but multi-state appliance using the proposed HHMM. We elaborate the modelling of the dishwasher (Kenmore 665.13242K900) shipped in the dataset with [21]. Although a dishwasher should normally be a multi-mode appliance, the author only published one profile so we also model it as a single-mode appliance here that has only one

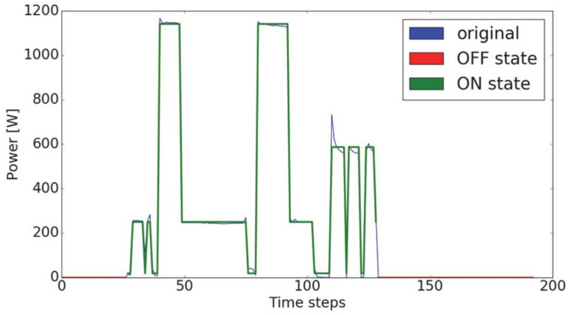


Fig. 5. The dishwasher power profile modelled by HHMM from [21].

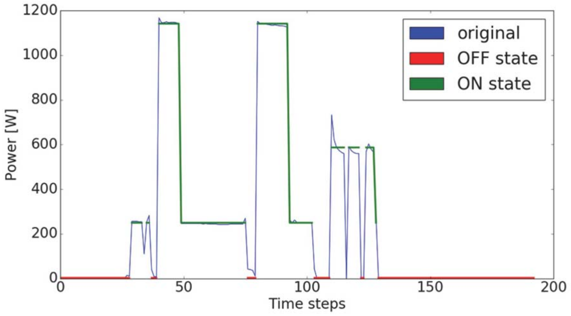


Fig. 6. The dishwasher power profile modelled by HMM from [21].

sub-HMM in the HHMM. Clearly, a dishwasher is a multi-state appliance. There are some standby periods when the dishwasher only consumes relatively small power during its active operation such as at time steps 75∼80, 105∼109, etc.

The proposed extended EM algorithm (Algorithm I) for HHMM is used for fitting this single-mode dishwasher. The input y is the down-sampled power sequence of the dishwasher. In this case K^U is 2: ON and OFF, since only one profile was given. $K^L = \{K_1^L,\ K_2^L\}$, where $K_1^L = 1$ because OFF state should be a single state that consumes zero power. K_2^L, on the other hand, is determined by the most widely used Bayesian information criterion (BIC) for the sub-HMM [22], [23]. The basic idea is to enumerate a number of dimension selections and perform the EM algorithm as given in Table I for model learning. Once finished, BIC scores are calculated and stored for all candidate models. The model with the minimum BIC score is then the selected model for the appliance. In the case of modelling the dishwasher, $K_2^L = 4$ according the BIC scores.

The fitting results of both the proposed HHMM and the conventional HMM are given in Fig. 5 and Fig. 6. The color of the lines distinguished the estimates between different upper states, i.e., ON and OFF. Red lines indicate the appliance is at the OFF state, while the green lines mean the appliance is active. The level of each piece-wise constant segment in the figures is determined by the most probable concrete state of the model, using the corresponding mean of its observation distribution $(y|X)$. It is shown that the HHMM is able to distinguish those standby states from the OFF states, while the conventional HMM considers them as the same state. In this case, the conventional HMM is more prone to equally attribute the next power change to the contribution of either the dishwasher or other devices, while the proposed HHMM puts more belief on the dishwasher as the cause of subsequent power levels.

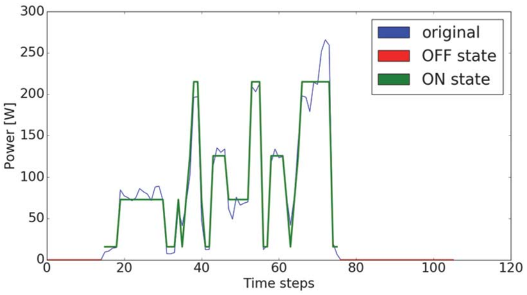


Fig. 7. Washing machine profile and the HHMM – normal wash cycle.

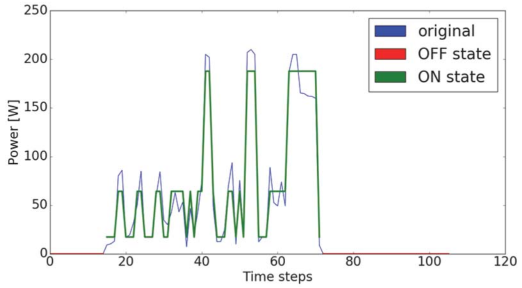


Fig. 8. Washing machine profile and the HHMM - permanent press cycle.

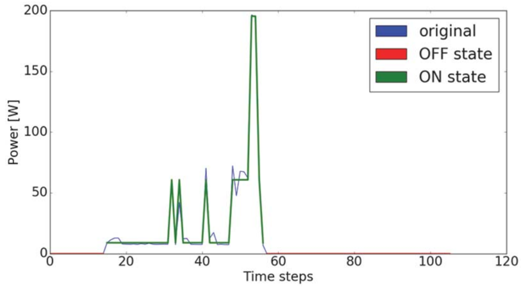


Fig. 9. Washing machine profile the HHMM - delicate cycle.

B. Fitting Multi-Mode Appliances

For a multi-mode washing machine (LG-WM2016CW1) with 3 distinct modes as illustrated by Fig. 7 - Fig. 9, the modelling process is similar to the single-mode dishwasher. The proposed Algorithm I is applied to each of the duty cycles respectively to yield three HHMMs, each of which has two upper states just like the dishwasher. Then, the three HHMMs are assembled into one HHMM because we know that these three modes belong to the same appliance. The assembling follows the 2 following rules: 1) without further information, we assume that all the modes of the appliance are equally possible to be used by the residents; 2) upon finishing any of the modes, the appliance should go back to the OFF mode before it can enter another mode. Therefore, for the assembled upper transition matrix, the self-transition probability of the OFF state is set to the mean of the OFF state self-transition probabilities of all three HHMMs for the washing machine,

TABLE II
SOME OF THE HHMM PARAMETERS FOR THE WASHING MACHINE

κ	μ_i	σ_i	$A^L_\kappa(i, end)$
OFF (state index 1)	0.00	1.00	0.022
Mode Normal (state index 2)	15.89	10.71	0.067
	72.95	10.43	0.000
	125.91	11.39	0.000
	215.11	25.92	0.000
Mode Permanent (state index 3)	17.2	7.32	0.032
	64.31	16.76	0.000
	187.65	20.40	0.000
Mode Delicate (state index 4)	8.92	2.44	0.045
	60.73	9.77	0.000
	195.43	1.08	0.000
Upper state transition matrix A^U	$\begin{matrix} 0.43 & 0.19 & 0.19 & 0.19 \\ 0.99 & 0.01 & 0 & 0 \\ 0.99 & 0 & 0.01 & 0 \\ 0.99 & 0 & 0 & 0.01 \end{matrix}$		
Mode Normal Lower State Transition Matrix $\tilde{A}^L_2$	$\begin{matrix} 0.57 & 0.29 & 0.14 & 0 \\ 0.17 & 0.70 & 0.09 & 0.04 \\ 0 & 0.20 & 0.60 & 0.20 \\ 0.15 & 0.08 & 0 & 0.77 \end{matrix}$		
Mode Permanent Lower State Transition Matrix $\tilde{A}^L_3$	$\begin{matrix} 0.53 & 0.43 & 0.04 \\ 0.36 & 0.55 & 0.09 \\ 0.15 & 0.08 & 0.77 \end{matrix}$		
Mode Delicate Lower State Transition Matrix $\tilde{A}^L_4$	$\begin{matrix} 0.87 & 0.13 & 0 \\ 0.44 & 0.44 & 0.12 \\ 0 & 0.50 & 0.50 \end{matrix}$		

while the probabilities to any modes are set to be equal. The self-transition probability of any mode other than OFF is set to a small value, with a large probability mass assigned to the transition to OFF mode.

The three fitted models are shown by the red lines (OFF states) and green lines (ON states) from Fig. 7 - Fig. 9. After assembling, the assembled HHMM parameters for the washing machine are detailed in Table II.

V. LOAD DISAGGREGATION WITH HHMM

A. Synthesis Test Case Design

The previous section has shown that the proposed HHMM can be learned via the EM framework. In this section, we are going to apply the learned models in the task of energy disaggregation.

The dataset in [21] includes typical duty cycles of 7 major appliance types. The summary of the dataset is given in Table III. In reality, the difficulty of disaggregation usually depends on how the profiles of different appliances are mixed at the same period of time. Because the power sequences in this dataset range from 30 minutes to several hours, we try to randomly mix a random set of 7 appliance types into a short 3-hour period, to generate sufficiently complicated test cases. The basic case generation rules are: 1) each of the 7 types of appliances should be switched on at a random time index of the first 2-hour period; 2) if a type has more than one appliance instances, a random one is picked; 3) if an appliance instance has more than one mode, a random profile of a mode is picked and placed at the random time index. The pseudo codes that we use to generate multiple test cases are given as Algorithm II in Table IV.

It should be pointed out that the Electric Stove (Kenmore_790.91312013) should be viewed as 4 appliances instead of an appliance with 4 modes. For simplicity of our test case design and the focus of verifying our proposed model, we treat it as an appliance with multi-modes just like the laundry dryer and the washing machine in the dataset. In this case, 3 out of 7 appliances in every test case are multi-mode appliances.

TABLE III
SUMMARY OF DATASET FROM [21]

Type	*Appliance*	*#Modes*	Identifier
Air Conditioning	Bryant_697CN030B	1	AC_Bry
	LG_LW1212ER	1	AC_LG
Electric Stove	Kenmore_790.91312013	4	ES_Ken
Dishwasher	Kenmore_665.13242K900	1	DW_Ken
Laundry Dryer	GE_WSM2420D3WW	3	LD_GE
	LG_DLE2516W	3	LD_LG
Refrigerator	Hotpoint_HTR16ABSRWW	1	RF_Hot
	Maytag_MSD2641KEW	1	RF_May
Washing Machine	GE_WSM2420D3WW	3	WM_GE
	LG_WM2016CW	3	WM_LG
Water Heater	E52_50R-045DV	1	WH_E52

TABLE IV
THE PSEUDO-CODES FOR TEST CASE GENERATION

```
Algorithm II: Generate Synthesis Test Cases
Inputs: dataset with 7 major types of appliances
Outputs: aggregate, ground_truth, appliance_info
Initialise a zero element matrix ground_truth with size (7, 10800)
For i = 1 to 7
    Pick an appliance of the type(i) randomly
    Pick a profile from all the modes of the appliance randomly
    Pick a timestep in a 2-hour range (1, 7200) to as the start time
    Record the by appliance_info.append(appliance, mode)
    Attach the profile, ground_truth(i, :) = profile
End For
Sum each timestep and yield the aggregate profile with size (1, 10800)
Down sample aggregate and ground_truth to 1 reading per minute
Return aggregate, ground_truth, appliance_info
```

In order to represent the varying complexity and diversity of real-life situations, we generated 30 test cases. For each test case, four scenarios are tested respectively and their results are compared. The four scenarios are: 1) appliance modes are modelled as single-mode HHMMs; 2) appliance modes are modelled as HMMs; 3) appliances are modelled as multi-mode HHMMs; 4) appliances are modelled as HMMs with concatenated mode profiles. In the first two scenarios, we assume that we have surveyed the previously unseen households and recorded not only the exact models of their major appliances, but also the exact modes of appliances they have only been using. In the Scenarios 3 and 4, we only assume we have known their models of appliances, but users have no fixed preference for the modes they are going to use. Our proposed HHMM is exactly designed to handle the Scenario 3. However, for HMM to model multi-mode appliances, a concatenated profile of all modes is the only way to do this. The learned HMM for the washing machine (LG-WM2016CW1) using concatenated profile is illustrated as Fig. 10. This is also the reason why other research requires lengthy power sequences to build appliance models before successful load disaggregation.

B. The Particle Filtering (PF) Solver

The particle filtering solver has been adopted to solve load disaggregation with HMM [12], [24]. One of the most notable

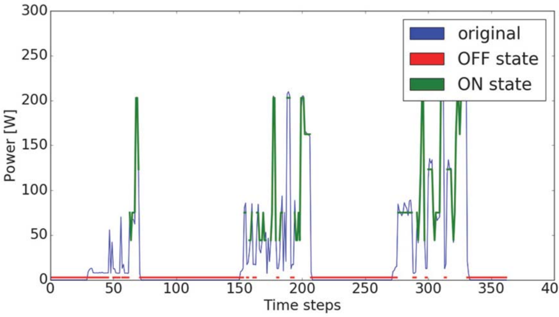


Fig. 10. The learned HMM for the 3 modes of the washing machine.

TABLE V
THE AVERAGE OF ESTIMATION ACCURACY OVER 20 TEST CASES

App. Identifier	S1	S2	S3	S4	# Picks
AC_Bry	0.7522	0.5411	0.7053	0.0843	15
AC_LG	0.8643	0.7966	0.8609	0.6428	15
ES_Ken	**0.6773**	**0.2085**	**0.6435**	**-0.4907**	**30**
DW_Ken	0.7402	0.5913	0.7363	0.3903	30
LD_GE	**0.7486**	**0.4412**	**0.7769**	**0.4823**	**14**
LD_LG	**0.5258**	**-0.0874**	**0.3984**	**0.0297**	**16**
RF_Hot	0.5491	0.2239	0.5284	0.2794	17
RF_May	0.6798	0.343	0.6922	0.3867	13
WM_GE	**0.6844**	**0.4066**	**0.694**	**0.2015**	**15**
WM_LG	**0.3918**	**0.0619**	**0.2483**	**-1.5722**	**15**
WH_E52	0.809	0.6612	0.7915	0.5321	30
Overall	0.7584	0.5194	0.7442	0.3137	30

pros of the PF solver is its linear scalability against the number of appliances. On the other hand, the most notable con of PF is that it is generally slow. We have built our own PF solver and demonstrated that it is much slower compared to the specialised optimisation based solver in [25].

However, due to the complexity of the proposed HHMM model, specialised solver with HHMM has not yet been developed. Because the better generality of the PF solver, in this paper, we extend the PF solver to make it able to work with the HHMM for load disaggregation, and demonstrate the effectiveness of our new appliance models by comparing results of the abovementioned four scenarios.

PF for HHMM should have an upper state sequence being sampled in time. The lower state transition is governed by the upper state. Specifically, the upper state for appliance n at time t of particle m is

$$x_t^{U(n)}[m] = \begin{cases} x_{t-1}^{U(n)}[m] & \text{if } f = 0 \\ \sim A^{U(n)}_{x_{t-1}^{U(n)}[m],*} & \text{if } f = 1 \end{cases} \tag{42}$$

where $\sim$ indicates the distribution of the random variable, m is the particle index, and $A_{i,*}^{U(n)}$ denotes the ith row of the upper state transition matrix for the nth appliance.

After the upper state of an appliance has been sampled, the lower state can be sampled as follows:

$$x_t^{L(n)}[m] \sim \begin{cases} \tilde{A}^{L(n)}_{x_{t-1}^{U(n)}[m]}\left(x_t^{L(n)}[m], *\right) & \text{if } f = 0 \\ \pi^{L(n)}_{\kappa x_{t-1}^{U(n)}[m]} & \text{if } f = 1 \end{cases} \tag{43}$$

where $\tilde{A}_{\kappa}^{L(n)}(i,\ *)$ denotes the i th row of the lower state transition matrix for the nth appliance when the upper state is κ.

The sampling of emission and the weight updating process is identical to PF for HMM, which was given in [25].

Assuming a load disaggregation problem which contains N appliances, T time steps, and each appliance has an average of K^U upper states and K^L lower states, the time complexity of running a PF for HHMM with M particles is $\mathcal{O}(MNT)$. At each iteration, there are 2 more variables being sampled, namely sampling the upper state and the finish flag compared to HMM. The complexity of PF is irrelevant to the number of appliance states. However, as the number of appliance states grows, the number of particles should be set to a larger value for the reliable results.

C. The Evaluation Metric

There are numerous evaluation metrics for load disaggregation. Makonin and Popowich [26] have done a very good review on many of them. Among all evaluation metrics, the estimation accuracy, firstly introduced by Kolter and Johnson [27], is by far the strictest metric of all. To be able to achieve full score (100%), one should correctly estimate not only the ON/OFF events of each appliance, but also the amount of power consumption at each time step along the problem length. The metric is calculated by:

$$Acc = 1 - \frac{\sum_{t=1}^{T}\sum_{n=1}^{N}\left|\hat{y}_t^{(n)} - y_t^{(n)}\right|}{2\sum_{t=1}^{T} y_t} \tag{44}$$

where $\hat{y}_t^{(n)}$ denotes the estimation of power consumption for the ith appliance at time t, and $y_t^{(n)}$ is the actual power consumption for the ith appliance.

D. Results and Comparison

We test the 30 generated cases under 4 different scenarios using the particle filtering solver. The number of particles is fixed to 5000. The results are summarised in Table V.

By comparing the performance in Scenarios 1 and 2, it is clearly seen that even the specific operating modes are known, modelling home appliances with HHMM achieves significant better performance than modelling the appliance consumption with conventional HMM.

Moreover, in the cases when the operating modes are unknown to the solver, such as in Scenarios 3 and 4, HHMM once again outperforms the conventional HMM by a huge margin. Compared to the scenarios where the modes are known, results with the HHMM modelling only have a slight decrease in estimate accuracy when the modes are unknown, while modelling multi-mode appliances with lengthy concatenated profiles sees a significant drop in the estimate accuracy when modes are known. This clearly demonstrates the contribution of our proposed model. With another layer to model the transition between appliance modes, the HHMM represents the real consumption behaviours of appliances better than the conventional HMM.

For the multi-mode appliances which are highlighted in bold characters in Table V, we can see the HHMM offers much better estimate. The accuracy of estimation for these appliances is

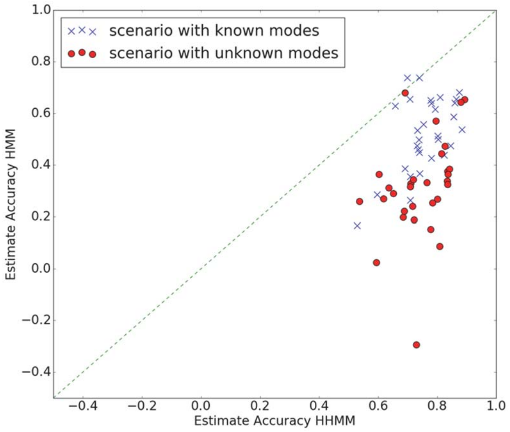


Fig. 11. Performance comparison between two models in different scenarios.

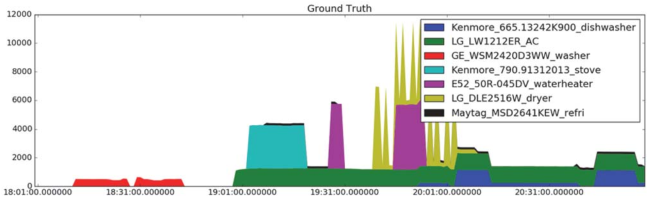


Fig. 12. The ground truth of Test Case No. 28.

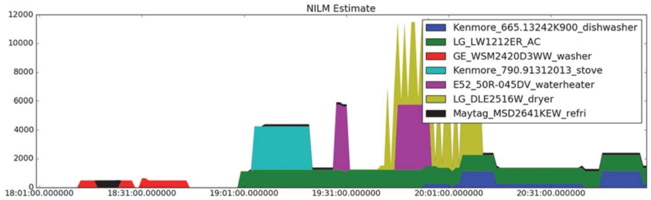


Fig. 13. The results of Scenario 1, accuracy = 0.8586.

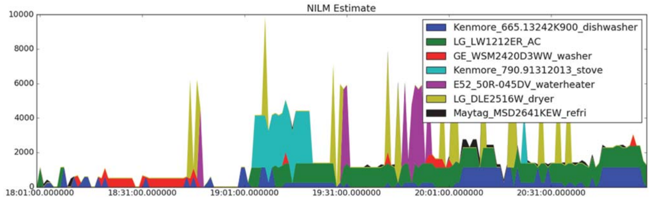


Fig. 14. The results of Scenario 2, accuracy = 0.6403.

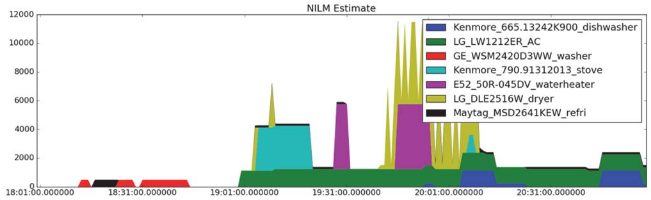


Fig. 15. The results of Scenario 3, accuracy = 0.8367.

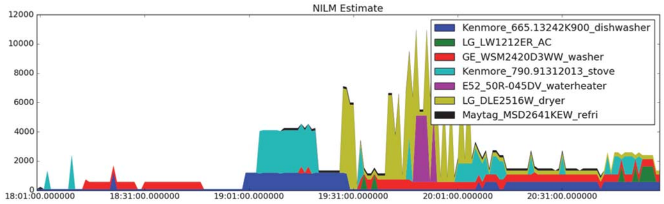


Fig. 16. The results of Scenario 4, accuracy = 0.3645.

TABLE VI
AVERAGE SOLVING TIME OF ALL SCENARIOS

Average Solving Time	S1	S2	S3	S4
Total seconds elapsed	2039	1826	1983	1845

TABLE VII
THE ESTIMATE ACCURACY COMPARISON IN REDD DATASET

Scenario	PF with HMM	SIQP + CP	PF with HHMM
6 appliances	0.6296	0.8161	0.7793
9 appliances	0.6252	0.5414	0.6069
12 appliances	0.5007	0.5286	0.5438

not affected by not knowing which modes were used under the HHMM scheme, while HMM completely fails in the scenarios without specific modes.

The overall estimate accuracy for the 30 test cases are given as Fig. 11. Despite the randomness of the generated problems, load disaggregation results based on HHMM are much more stable than those based on HMM. Under the scheme of HHMM, the results range from 0.53 to 0.89, while the results range from -0.29 to 0.68. A negative value in the results means that some power of some appliances is incorrectly assigned to the wrong appliances completely.

In order to demonstrate more clearly the benefit of the HHMM, Fig. 12 to Fig. 16 visualise the load disaggregation results of the Test Case No. 9 for all 4 scenarios. In this fully mixed problem, it is clearly shown that in Scenarios 2 and 4, there are more false positive time steps than the scenarios with HHMM. This is the benefit of the HHMM's capability of distinguishing standby states from OFF states.

In terms of solving efficiency, Table VI confirms that the solution time of particle filtering solver is irrelevant against the number of appliance states. Because the two more variables to be sampled at each iteration, the PF with HHMM generally takes slightly longer to process. Solution time for a 3-hour problem is just about half an hour. Considering the linear complexity against problem length T, our proposed model can work with actual smart meter infrastructure, because it takes about 10 minutes to solve an hourly long problem and smart meters communicate with utilities once per hour.

E. Real-World Test Cases: REDD Dataset

In order to further justify our proposed model, we also apply the PF with HHMM framework to the most popular load disaggregation dataset – REDD dataset [27]. Since the REDD dataset does not provide detailed information about appliance modes, we extract the first duty cycle of each appliance and model them as single-mode appliances using HHMM. In order to compare with the previous work, we also test the our new models in the 3 different scenarios as described in [25], with 6, 9 and 12 appliances respectively.

The results and their comparison are summarised in Table VII.

The test results on real-world dataset also confirm the potential of our hierarchical model. Although the performance decreases as the number of appliances increases, one of the causes is that most appliances in real-world practice have multiple modes but they are modelled as single-mode appliances in our real-world tests.

VI. Conclusion

This paper is inspired by the area of speech recognition and proposes to apply the HHMM to model home appliances so that features of multi-mode appliances can be captured. A more concise and consistent EM framework for HHMM model fitting is derived. Comprehensive tests on public datasets have verified that the extra layer of mode transition in the HHMM can help to stabilise the ON/OFF transition even for single-mode appliances, which leads to a much better result in load disaggregation problems.

In future, the proposed HHMM can be further extended to combine with other sophisticated models such as hidden semi-Markov model (HSMM) [11], conditional HSMM [11], difference HMM [14], and the difference HSMM [15]. The HHMM can enable the ability to distinguish standby states from OFF states for these models.

Also, future research on more efficient solvers with HHMM is required. Although the current PF with HHMM solving framework is compatible with the smart meter infrastructure, more efficient specialised algorithms can maximise the potential of the HHMM in future practical use.

References

[1] M. Kavgic *et al.*, “A review of bottom-up building stock models for energy consumption in the residential sector,” *Build. Environ.*, vol. 45, no. 7, pp. 1683–1697, 2010.

[2] A. Chrysopoulos, C. Diou, A. L. Symeonidis, and P. A. Mitkas, “Bottom-up modeling of small-scale energy consumers for effective demand response applications,” *Eng. Appl. Artif. Intell.*, vol. 35, pp. 299–315, Oct. 2014.

[3] G. W. Hart, “Nonintrusive appliance load monitoring,” *Proc. IEEE*, vol. 80, no. 12, pp. 1870–1891, Dec. 1992.

[4] L. Farinaccio and R. Zmeureanu, “Using a pattern recognition approach to disaggregate the total electricity consumption in a house into the major end-uses,” *Energy Build.*, vol. 30, no. 3, pp. 245–259, Aug. 1999.

[5] D. Srinivasan, W. S. Ng, and A. C. Liew, “Neural-network-based signature recognition for harmonic source identification,” *IEEE Trans. Power Del.*, vol. 21, no. 1, pp. 398–405, Jan. 2006.

[6] A. Reinhardt, D. Christin, and S. S. Kanhere, “Predicting the power consumption of electric appliances through time series pattern matching,” in *Proc. 5th ACM Workshop Embedded Syst. Energy Efficient Build.*, Rome, Italy, 2013, pp. 1–2.

[7] M. Dong, P. C. M. Meira, W. Xu, and C. Y. Chung, “Non-intrusive signature extraction for major residential loads,” *IEEE Trans. Smart Grid*, vol. 4, no. 3, pp. 1421–1430, Sep. 2013.

[8] T. Hassan, F. Javed, and N. Arshad, “An empirical investigation of V-I trajectory based load signatures for non-intrusive load monitoring,” *IEEE Trans. Smart Grid*, vol. 5, no. 2, pp. 870–878, Mar. 2014.

[9] Y.-H. Lin and M.-S. Tsai, “An advanced home energy management system facilitated by nonintrusive load monitoring with automated multiobjective power scheduling,” *IEEE Trans. Smart Grid*, vol. 6, no. 4, pp. 1839–1851, Jul. 2015.

[10] NSMP Business Requirements Work Group. (2011). *Smart Metering Infrastructure Minimum Functionality Specification*. [Online]. Available: https://link.aemo.com.au/sites/wcl/smartmetering/Pages/BRWG.aspx

[11] H. Kim, M. Marwah, M. F. Arlitt, G. Lyon, and J. Han, “Unsupervised disaggregation of low frequency power measurements,” in *Proc. SIAM Int. Conf. Data Mining*, Mesa, AZ, USA, 2011, pp. 747–758.

[12] D. Egarter, V. P. Bhuvana, and W. Elmenreich, “PALDi: Online load disaggregation via particle filtering,” *IEEE Trans. Instrum. Meas.*, vol. 64, no. 2, pp. 467–477, Feb. 2015.

[13] S. Makonin, F. Popowich, I. V. Bajiæ, B. Gill, and L. Bartram, “Exploiting HMM sparsity to perform online real-time nonintrusive load monitoring,” *IEEE Trans. Smart Grid*, vol. 7, no. 6, pp. 2575–2585, Nov. 2016.

[14] J. Z. Kolter and T. Jaakkola, “Approximate inference in additive factorial HMMs with application to energy disaggregation,” in *Proc. 15th Int. Conf. Artif. Intell. Stat.*, La Palma, Spain, 2012, pp. 1472–1482.

[15] Z. Guo, Z. J. Wang, and A. Kashani, “Home appliance load modeling from aggregated smart meter data,” *IEEE Trans. Power Syst.*, vol. 30, no. 1, pp. 254–262, Jan. 2015.

[16] S. Fine, Y. Singer, and N. Tishby, “The hierarchical hidden Markov model: Analysis and applications,” *Mach. Learn.*, vol. 32, no. 1, pp. 41–62, 1998.

[17] K. P. Murphy, “Dynamic Bayesian networks: Representation, inference and learning,” Ph.D. dissertation, Dept. Comput. Sci., Univ. California at Berkeley, Berkeley, CA, USA, 2002.

[18] T. Rydén, “EM versus Markov chain Monte Carlo for estimation of hidden Markov models: A computational perspective,” *Bayesian Anal.*, vol. 3, pp. 659–688, Nov. 2008.

[19] O. Parson, S. Ghosh, M. Weal, and A. Rogers, “Non-intrusive load monitoring using prior models of general appliance types,” in *Proc. 26th Conf. Artif. Intell. (AAAI)*, Toronto, ON, Canada, 2012, pp. 356–362.

[20] J. Kelly and W. Knottenbelt, “Neural NILM: Deep neural networks applied to energy disaggregation,” in *Proc. 2nd ACM Int. Conf. Embedded Syst. Energy Efficient Built Environ.*, Seoul, South Korea, 2015, pp. 55–64.

[21] M. Pipattanasomporn, M. Kuzlu, S. Rahman, and Y. Teklu, “Load profiles of selected major household appliances and their demand response opportunities,” *IEEE Trans. Smart Grid*, vol. 5, no. 2, pp. 742–750, Mar. 2014.

[22] Z. Lu and A. M. Zoubir, “Generalized Bayesian information criterion for source enumeration in array processing,” *IEEE Trans. Signal Process.*, vol. 61, no. 6, pp. 1470–1480, Mar. 2013.

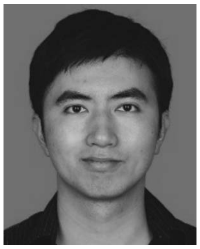
[23] G. Schwarz, “Estimating the dimension of a model,” *Ann. Stat.*, vol. 6, pp. 461–464, Nov. 1978.

[24] Y. Wong, T. Drummond, and Y. A. Sekercioglu, “Real-time load disaggregation algorithm using particle-based distribution truncation with state occupancy model,” *Electron Lett.*, vol. 50, no. 9, pp. 697–699, Apr. 2014.

[25] W. Kong, Z. Y. Dong, D. J. Hill, F. Luo, and Y. Xu, “Improving nonintrusive load monitoring efficiency via a hybrid programming method,” *IEEE Trans. Ind. Informat.*, vol. PP, pp. 1–1, 2016.

[26] S. Makonin and F. Popowich, “Nonintrusive load monitoring (NILM) performance evaluation,” *Energy Efficiency*, vol. 8, no. 4, pp. 809–814, 2015.

[27] Z. J. Kolter and M. J. Johnson, “REDD: A public data set for energy disaggregation research,” in *Proc. Data Min. Appl. Sustain.*, vol. 25. San Diego, CA, USA, 2011, pp. 59–62.



Weicong Kong (S’14) received the B.E. and M.E. degrees from the South China University of Technology, Guangzhou, China, in 2008 and 2011, respectively, and the M.Sc. degree from the University of Strathclyde, Glasgow, U.K., in 2009. He is currently pursuing the Ph.D. degree with the School of Electrical and Information Engineering, University of Sydney. He was the Electrical Engineer with Shenzhen Power Supply Company, where he is in charge of the development of distribution automation system, SCADA, and AMI. His research interests include nonintrusive load monitoring, smart grid, probabilistic graphical models, machine learning, and demand response.

Zhao Yang Dong (M'99–SM'06) received the Ph.D. degree from the University of Sydney, Australia, in 1999, where he is currently a Professor and the Head of the School of Electrical and Information Engineering. He was the Ausgrid Chair and the Director of the Ausgrid Centre of Excellence for Intelligent Electricity Networks, University of Newcastle, Australia. He also held academic and industrial positions with Hong Kong Polytechnic University and Transend Networks (currently, TASNetworks), TAS, Australia. His research interest includes smart grid, power system planning, power system security, load modeling, renewable energy systems, electricity market, and computational intelligence and its application in power engineering. He is an Editor of the IEEE TRANSACTIONS ON SMART GRID, the IEEE TRANSACTIONS ON SUSTAINABLE ENERGY, the IEEE POWER ENGINEERING LETTERS, and *IET Renewable Energy Generation*.

David J. Hill (S'72–M'76–SM'91–F'93–LF'14) received the Ph.D. degree in electrical engineering from the University of Newcastle, Australia, in 1976. He holds the Chair of Electrical Engineering with the Department of Electrical and Electronic Engineering, University of Hong Kong. He is also a part-time Professor of Electrical Engineering with the University of Sydney, Australia.

From 2005 to 2010, he was an Australian Research Council Federation Fellow with the Australian National University. Since 1994, he has held various positions with the University of Sydney, Australia, including the Chair of Electrical Engineering until 2002 and from 2010 to 2013 along with an ARC Professorial Fellowship. He has also held academic and substantial visiting positions with the University of Melbourne, the University of California, Berkeley, the University of Newcastle, Australia, the University of Lund, Sweden, the University of Munich, the City University of Hong Kong, and Hong Kong Polytechnic University.

Dr. Hill's general research interests are in control systems, complex networks, power systems and stability analysis. His work is mainly on control and planning of future energy networks and basic stability and control questions for dynamic networks.

Prof. Hill is a fellow of the Society for Industrial and Applied Mathematics, USA, the Australian Academy of Science and the Australian Academy of Technological Sciences and Engineering. He is also a Foreign Member of the Royal Swedish Academy of Engineering Sciences.

J. Ma, photograph and biography not available at the time of publication.

J. H. Zhao, photograph and biography not available at the time of publication.

F. J. Luo, photograph and biography not available at the time of publication.