Real-time load disaggregation algorithm using particle-based distribution truncation with state occupancy model

Yung Fei Wong, T. Drummond and Y.A. Şekercioglu

A new particle-based distribution truncation method with a duration-dependent hidden semi-Markov model for non-intrusive appliance load monitoring is presented. Unlike earlier works, the approach keeps track of a set of states without prematurely pruning away intermediate ranked states. It also enables appliance state duration characteristics to be incorporated in a straightforward manner. Results show that the approach outperforms both the Viterbi algorithm and conventional particle-filtering methods.

Introduction: Non-intrusive appliance load monitoring (NIALM) uses aggregate information to mathematically infer the power consumption of contributing appliances [1]. Past works in the field typically employed methods to find the best point estimates at each time step. Prime examples include the maximum a posteriori probability [2] and the Viterbi algorithm [3] which only keep the best previous state for each possible current state value. This is undesirable because the most probable previous state does not necessarily correspond to the true state. In addition, measurements obtained so far may not provide sufficient information for good decisions to be made. Very recently, particle-filtering (PF) with a factorial hidden Markov model (FHMM) for appliance state tracking was proposed [4]. However, PF is encumbered by the problem of low particle diversity [5]. In the case of NIALM, where a given observed aggregate signal can be explained by many different combinations of component signals, this problem is wasteful considering that particle slots are premium and identical off-springs (with similar state sequence) offer no benefits but take up particle slots that could otherwise be occupied by particles with different state sequences. In addition, the use of FHMM implicitly assumes that durations of all appliance states are geometrically distributed, which in reality is clearly not the case.

In this Letter, we propose a new particle-based distribution truncation (PDT) method with a duration-dependent (DD) state transition model which does away with such limitations. Our model is inspired by Vaeghe [6], and unlike typical hidden semi-Markov models [7] our formulation allows appliance state duration characteristics to be easily incorporated into the state transition model. The resulting inference stage is computationally efficient with no explicit search over all possible state durations.

Tests were performed on a public dataset of real homes [8]. Results show that our approach performs better than the Viterbi algorithm and the PF method as outlined in [4].

Problem statement: Given the aggregate power measurement, \( O_t \), the goal is to estimate the state of each appliance in the system. In a formal sense, we would like to infer the state \( S_{tk} \) of an appliance \( k \) at time \( t \) (where \( S_{tk} \in \{0, \ldots, m_k - 1\} \) and \( m_k \) is the number of possible states) such that

\[
O_t = \sum_k f(k) S_{tk}
\]  

(1)

where \( f(k) \) is a function mapping \( S_{tk} \) to the power consumption of appliance \( k \).

System model: We model the problem using the dynamic Bayesian network (DBN) as shown in Fig. 1. \( C_t \) refers to how long the system has dwelt in the current state \( S_t \). It should be noted that both \( S_t \) and \( C_t \) are vectors with elements corresponding to the state of each appliance and how long each appliance has stayed in its current state, respectively. The model uses the fact that appliances are more likely to switch states, the longer they stay in their current state. This is contrary to FHMM [4] which says that the probability of switching states is the same regardless of how long a given appliance has been in its current state. As such, FHMM is a poor model for appliance behaviour.

A significant advantage of our model is that it allows inference algorithms to discern between different appliances with similar power consumption. If we consider the recursive expression of \( P(S_{tk}, O_t) \):

\[
P(S_{tk}, O_t) = P(S_{tk-1}, O_{t-1})P(O_t|S_{tk})P(S_{tk}|C_{tk-1})(2)
\]

the third term on the right, \( P(S_{tk}|C_{tk-1}) \) essentially modulates \( P(O_t|S_t) \) which means any sort of shared characteristics for \( P(O_t|S_t) \) between different states can be resolved.

The DBN shown in Fig. 1 is characterised by the following equations:

\[
P(S_{tk}|C_{tk}) = \prod_k P(S_{tk}|S_{tk-1}, C_{tk-1})
\]

(3)

\[
P(C_{tk}|C_{tk-1}, S_{tk-1}, S_t) = \begin{cases} \delta_{S_{tk}, S_{tk-1}} & \text{if } S_{tk} = S_{tk-1} \\ \delta_{1}(C_{tk}) & \text{otherwise} \end{cases}
\]

(4)

\[
P(O_t|S_t) = N\left(\sum_{k=1}^{K} \mu_{S_{tk}}, \sum_{k=1}^{K} \sigma_{S_{tk}}^2\right)
\]

(5)

where \( \delta \) and \( K \) denote the Kronecker delta function and the number of appliances in the system, respectively.

Each product term on the right-hand side of (3) can be derived from the state duration distribution \( P(T_{tk}) \). For appliance \( k \), it can be shown that

\[
P(S_{tk} \neq 0, S_{tk} = m_s S_{tk-1} = l, C_{tk-1}) = \frac{P(T_{tk} = C_{tk-1}) P(S_{tk} = m_s S_{tk-1} = l)}{P(T_{tk} \geq C_{tk-1})}
\]

(6)

and

\[
P(S_{tk} = l|S_{tk-1} = l, C_{tk-1}) = 1 - \frac{P(T_{tk} = C_{tk-1})}{P(T_{tk} \geq C_{tk-1})}
\]

(7)

As an example, Fig. 2 shows the DD state transition probabilities from state 0 for the fridge of House 2 of the REDD dataset [8], derived from each of the fridge’s state duration distribution. It can be seen that the probability of staying in state 0 decreases as the dwell time of state 0 increases. This relationship is consistent with our intuition.

Fig. 2 Fridge’s DD state transition probabilities (from state 0)

The Gaussian parameters for the emission probability, as shown in (5), are obtained using maximum-likelihood estimation. Although only real power is used as an observation feature for our case, \( O_t \) can be made to represent more complex features such as current harmonics if more informative datasets are available. Doing so may further improve disaggregation performance.

Appliance state inference: Our PDT algorithm works by keeping track of \( N \) best ‘unique’ state sequences (\( N \) particles) at each time step. Fig. 3 illustrates the process in which we first consider all \( M \) states in the full state-space. As there are naturally only a few states which satisfy the constraint of (1), distribution \( P(S_t|O_t) \) will be sparse. This sparsity is
exploited to reduce computational cost by keeping only those states for which $P(S_t | O_t) > 0$. If we denote $L$ as the size of the reduced set, $L$ will be such that $L \ll M$.

$$s_1 \rightarrow \ldots \rightarrow s_M$$

![Mechanics of algorithm](image)

**Fig. 3** Mechanics of algorithm

The reduced set of states is next concatenated with the $N$ best particles for time $t-1$ to give a temporary expanded list of $NL$ particles. Subsequently, each particle in the expanded list is scored using $P(S_t | O_t)$ before being sorted in descending order, de-duplicated and truncated to produce a new set of $N$ particles for time $t$. The benefits of our approach are twofold. First, the sorting step allows the possibility of old particles in previous time steps to switch from an immediately ranked position to a highly ranked position if it is determined that they are better at jointly explaining the old and new observations. Secondly, the de-duplication step guarantees high particle diversity and prevents more particle slots such that more immediately ranked particles are kept.

**Disaggregation performance:** Our algorithm was tested and trained on House 2 data of the REDD dataset [8]. The first two-thirds of the data are used for training, whereas the remaining one-third forms the test set. The dataset consists of nine appliances, each monitored for approximately one month. However, two appliances – washer-dryer and dishwasher – are not monitored and disposed do not have sufficient ON-state data to be included in the analysis. They are thus removed from consideration. For evaluating the disaggregation performance, both precision and recall [9] are used as metrics. Table 1 shows the disaggregation performance of our approach (5000 particles) relative to the Viterbi algorithm and the PF method (5000 particles).

**Table 1:** Disaggregation performance

<table>
<thead>
<tr>
<th>Appliance</th>
<th>Precision (%)</th>
<th>recall (%)</th>
<th>PDT</th>
<th>Viterbi algorithm</th>
<th>PF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kitchen outlets 1</td>
<td>82.9/71.3</td>
<td>43.0/64.9</td>
<td>80.2/66.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lighting</td>
<td>69.1/82.9</td>
<td>23.4/78.6</td>
<td>82.8/83.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stove</td>
<td>19.1/95.4</td>
<td>5.2/96.4</td>
<td>29.6/95.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Microwave</td>
<td>92.4/92.9</td>
<td>59.1/87.4</td>
<td>84.7/79.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kitchen outlets 2</td>
<td>97.3/96.9</td>
<td>93.7/81.5</td>
<td>30.4/42.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Refrigerator</td>
<td>96.3/91.8</td>
<td>86.3/64.4</td>
<td>95.8/93.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dishwasher</td>
<td>80.7/95.1</td>
<td>63.1/72.5</td>
<td>34.5/34.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>76.8/89.3</td>
<td>53.4/71.1</td>
<td>62.5/70.8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The Viterbi algorithm and the PF method suffer from the same underlying problem through direct means and indirect means, respectively. That is, immediately ranked inferences are pruned without due consideration that one of the said inferences may be the true state in light of future observations. Specifically, in the case of the Viterbi algorithm, only the most probable previous state leading to each current state is chosen. Whereas in the PF case, a given inference may occupy many particle slots and the indirect consequence is that only a small number of immediately ranked inferences are kept. Our particle truncation approach addresses these issues and it has been shown to improve disaggregation performance. Fig. 4 illustrates the estimated energy breakdown obtained using our method as compared to the true energy breakdown.

![Actual energy breakdown](image)

**Fig. 4** Actual energy breakdown against estimated energy breakdown

In terms of runtime efficiency, our algorithm (using 5000 particles) takes an average of 0.09 s to process one sample. This is with a MATLAB implementation running on an Intel Core i7-2600 personal computer with 8 GB of random access memory. Further improvement in processing speed should be apparent when implemented on C. Having said that, it may be of interest to consider the number of particles. Theoretically, increasing the number of particles would improve disaggregation performance at the expense of computational efficiency. This can be seen as a performance–efficiency trade-off. Real-world implementations may want to take into account the desired disaggregation performance, the implementation platform memory capacity and the optimal number of particles.

**Conclusion:** We have devised a new PDT method for real-time tracking of home appliances utilising only aggregate real power measurements. Appliance state duration characteristics are explicitly modelled and indirectly accounted for using the derived expression of the DD state transition probabilities. The latter enables efficient computation of $P(S_t | O_t)$ without looping over all possible state durations. In addition, the inference step guarantees the uniqueness of particles, which allows the production of a diverse set of particles at each time step, be it highly ranked or immediately ranked. Overall, the results indicate that our approach yields excellent disaggregation performance as compared to the Viterbi algorithm and conventional PF methods.

References