The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Model-Free Iterative Temporal Appliance Discovery for Unsupervised Electricity Disaggregation

Mark Valovage, Akshay Shekhawat, Maria Gini
Computer Science and Engineering
University of Minnesota, Minneapolis, MN
{valov002,shekh027,gini}@umn.edu

Abstract

Electricity disaggregation identifies individual appliances from one or more aggregate data streams and has immense potential to reduce residential and commercial electrical waste. Since supervised learning methods rely on meticulously labeled training samples that are expensive to obtain, unsupervised methods show the most promise for widespread application. However, unsupervised learning methods previously applied to electricity disaggregation suffer from critical limitations. This paper introduces the concept of iterative appliance discovery, a novel unsupervised disaggregation method that progressively identifies the 'easiest to find' or 'most likely' appliances first. Once these simpler appliances have been identified, the computational complexity of the search space can be significantly reduced, enabling iterative discovery to identify more complex appliances. We test iterative appliance discovery against an existing competitive unsupervised method using two publicly available datasets. Results using different sampling rates show iterative discovery has faster runtimes and produces better accuracy. Furthermore, iterative discovery does not require prior knowledge of appliance characteristics and demonstrates unprecedented scalability to identify long, overlapped sequences that other unsupervised learning algorithms cannot.

Introduction

Every year, wasted energy costs the United States $130 billion and produces over a gigaton of pollution, over half of which comes from wasted electricity (Granade et al. 2009). A variety of Artificial Intelligence applications have been proposed to reduce waste, including automated energy pricing and demand response (Malik and Lehtonen 2016), but some of the largest potential savings require adjustments to consumer behavior. Behavioral adjustments that have a minimal effect on consumers' lifestyles could reduce residential consumption by up to 20%, but applications to drive these adjustments remain largely unimplemented due to limitations in existing approaches (Frankel, Heck, and Tai 2013). Electricity disaggregation is one such application.

Electricity disaggregation, also called Non-Intrusive Appliance Load Monitoring or Single Point Sensing (Gupta, Reynolds, and Patel 2010), is the process of extracting energy usage for multiple appliances from a single aggregate electrical signal (Hart 1992). While existing smart meters have the capability to monitor the consumption of individual appliances, the cost of purchasing a smart meter for every appliance in a building greatly outweighs any potential savings obtained by reducing waste. In contrast, electricity disaggregation can monitor energy usage of all appliances with a single smart meter (Carrie Armel et al. 2012).

Appliance-specific data has multiple uses. First, reporting appliance-specific power consumption to consumers has a measurable impact on reducing waste. Second, an automated system could use such data to provide recommendations to consumers on specific actions to reduce waste, identify outdated appliances and recommend efficient replacements, and (if given authority) could automatically turn off unused appliances or shift their operation to times when electricity is cheaper. Third, real-world data could help manufacturers improve efficiency of future appliance models. Finally, utility companies could use such data for better market segmentation and load forecasting, reducing operational costs by up to 28% (Lobaccaro, Carlucci, and Löfström 2016).

Despite significant research in electricity disaggregation, existing methods remain unimplemented in the real world. Supervised learning requires multiple isolated training samples for each appliance, a time-consuming task most consumers are unwilling to perform. In contrast, unsupervised learning methods have the potential to disaggregate appliances without training samples, but their ability to discover appliances is limited since they can only discover appliances that happen to occur in isolation, require pre-existing appliance models, or use brute force search and are restricted to discovering appliances with very short sequence lengths.

Contributions: This paper introduces iterative appliance discovery, an unsupervised disaggregation technique that makes no prior assumptions on appliances. Iterative discovery reconstructs appliances from detected events by identifying the simplest appliances with the closest temporal events first. Following each new discovery, iterative discovery reduces the search space complexity, enabling identification of additional, more complex appliances. This novel approach yields higher accuracy, faster execution time, and scalability to longer event sequences than any previously introduced unsupervised disaggregation algorithm. We demonstrate these improvements on two publicly available datasets containing 7 houses against an existing competitive method.

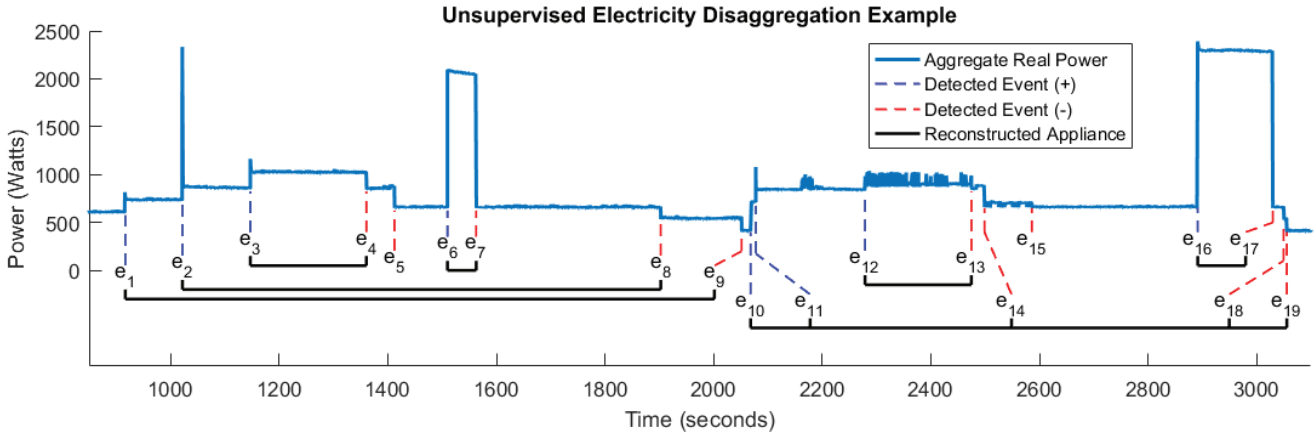

Figure 1: Example of disaggregation from the BLUED dataset spanning 35 minutes. Aggregate power (dark blue) is first parsed into events (dotted lines; power increases in blue, decreases in red). Valid episodes from appliances are shown in black. Most unsupervised methods can discover non-overlapped episodes such as (e_3,e_4) and (e_{16},e_{17}), while others can find short overlapping episodes, such as (e_2,e_8) and (e_1,e_9). Only iterative discovery can find longer overlapped episodes like (e_{10},e_{11},e_{14},e_{18},e_{19}).

Related Work

Supervised learning models applied to electricity disaggregation include Discriminative Sparse Coding (Kolter, Batra, and Ng 2010), 1-Nearest Neighbor (Gupta, Reynolds, and Patel 2010), Semi-Definite Program Relaxation with Randomized Rounding (Shaloudegi et al. 2016), Markov random fields (Tomkins, Pujara, and Getoor 2017), and others (Zoha et al. 2012). However, supervised learning methods have limited application in real-world settings, since they require labeled samples for training or separate meters for each appliance, both of which are expensive to obtain.

Unsupervised disaggregation shows more promise in real-world implementation (Zoha et al. 2012), but existing methods have different limitations. AFAMAP (Kolter and Jaakkola 2012) requires at least one instance of an appliance to be observed in isolation, meaning appliances that never occur in isolation will never be discovered. Temporal Motif Mining (Shao, Marwah, and Ramakrishnan 2013) limits event sequence lengths to 3 or less, restricting discovery to simple appliances under ideal sampling conditions. (Parson et al. 2012) proposed an iterative approach to appliance discovery, but it requires initial models of appliance types and can only disaggregate a few appliances with large signatures.

A Review of Unsupervised Disaggregation

Unsupervised disaggregation consists of at least two distinct steps: *event detection* and *episode discovery*. *Event detection* first segments the aggregate power data stream into significant events. Each event corresponds to an appliance state transition, such as a light turning on or off, or a TV changing its brightness. *Episode discovery* then uses these detected events to discover appliances. Figure 1 shows an example.

While the focus of this paper is on episode discovery, it is worth noting that detecting events from real-world appliances is a nontrivial problem. Figure 1 spans 35 minutes, meaning all events may appear abrupt at first glance due to the scale. However, each event has a different shape and different duration before power draw becomes steady again. Background noise also varies significantly. Some states in Figure 1 have minimal noise, while others have large variations in noise, such as the interval between events e_{12} and e_{13}. Finally, the time between events also varies, amplifying the challenge of detecting events. Event detection methods are further detailed in the Experimental Setup section, and we focus the remainder of the paper on episode discovery.

Definitions

Given an aggregate time sequence T of real power observations, an *event* is a timestamp $e_i \in \mathbb{R}^+$ identifying a state change in the aggregate power data stream. The sequence of detected events $e_1, e_2, ..., e_n$ for T is chronologically ordered. The sequence $p_1, p_2, ..., p_n$ details the associated changes in real power for each event, where $p_i \in \mathbb{R}$ denotes each change in real power associated with event e_i.

An *episode* $E = (e_{n_1}, e_{n_2}, ..., e_{n_L})$ is a short subsequence of detected events, where $n_1, n_2, ..., n_L$ is a strictly increasing sequence in $\mathbb{N}$. This is similar to the data mining definition (Mannila, Toivonen, and Verkamo 1995; 1997), but in this context an episode must be a *serial* episode, meaning events occur sequentially in a specific order.

An *appliance cycle* is a set of state changes for a single appliance where the appliance's power draw begins and ends at 0 watts. A *valid episode* is one that has been validated through axiomatic constraints (detailed later) to be an appliance cycle. A *candidate episode* is an episode that has not been tested to see if it is valid; in other words, it may or may not be valid sequence of events from an appliance cycle.

Search Complexity and Existing Limitations

Given a sequence of detected events $e_1, e_2, ..., e_n$, the goal in unsupervised disaggregation is to reconstruct appliances from these events. Previous attempts have been limited to appliances that happen to occur entirely in isolation (Kolter and Jaakkola 2012) or use brute force search to explore pos-

sible episodes and cannot identify episodes with more than 3 events (Shao, Marwah, and Ramakrishnan 2013).

For a sequence of detected events of length n, there are $\binom{n}{L}$ possible subsequences of length L that preserve the order of the original sequence. Generally, $L << n$, since appliance event sequences tend to be short. Even so, there is an exponential increase in the number of candidate episodes as L increases. While the 19 events in Figure 1 are a relatively small number (Figure 1 spans only 35 minutes), there are $\binom{19}{2}$ = 171 candidate episodes of length 2, $\binom{19}{3} \approxeq$ 1,000 candidates of length 3, and $\binom{19}{5} > 11{,}000$ candidates of length 5. Often the number of detected events will be much larger.

Furthermore, computing resources in electricity disaggregation are extremely limited. To reduce waste, any disaggregation algorithm must run in real time on inexpensive, low power hardware. This leads us to iterative discovery.

Iterative Episode Discovery

Approach and Key Concepts

Iterative appliance discovery, which we will refer to interchangeably as *iterative episode discovery*, is based on two intuitive concepts that are used to intelligently generate candidate episodes and avoid brute force search complexities.

First, the difficulty of discovering an appliance depends on its complexity. Simpler appliances are easier to discover than complex ones. Type I appliances, such as lights and toasters, are the simplest since they only have 2 states: ON and OFF. More complex type II appliances have multiple discrete states, such as a pump or fan with LOW, MEDIUM, and HIGH settings. Type III appliances such as dimmer switches, TVs, and computers have a continuous number of states and are the hardest to discover (Zoha et al. 2012).

While our approach does not rely on these different types of appliances, it is natural to observe that identifying simpler appliances first is easier than identifying complex appliances. Simpler appliances tend to be shorter in duration and have less variation in power changes than complex ones.

Second, events that are temporally close are more likely to come from the same appliance than events that occur far apart. In the ideal case, all of the events associated with an episode occur in isolation, such as (e_3, e_4) and (e_{16},e_{17}) in Figure 1. In general, ($e_i, e_{i+1}, ..., e_{i+L-1}$) represents an isolated episode of length L starting at event e_i.

However, an appliance's operation will often not be isolated and will overlap with other appliances. Given a valid episode from a single appliance, we refer to events produced by other appliances as *external events*. For example, the episode (e_{10},e_{11},e_{14},e_{18},e_{19}) in Figure 1 has 5 external events: e_{12}, e_{13}, e_{15}, e_{16}, and e_{17}. Episodes with fewer external events are combinatorially easier to discover, and, in general, are more likely to come from a single appliance.

Axiomatic Constraints for Episode Validation

To test for validity, we adopt two axioms from (Shao, Marwah, and Ramakrishnan 2013) that are physical requirements for any appliance cycle. The axioms below test a candidate episode $E = (e_{n_1}, e_{n_2}, ..., e_{n_L})$ of length L with associated power changes $P = (p_{n_1}, p_{n_2}, ..., p_{n_L})$. Here, $n_1, n_2, ..., n_L$ is a strictly increasing sequence of natural numbers denoting the events in the candidate episode.

Axiom 1 (ϕ_1): Conservation of Power. Since the power draw for any appliance cycle must begin and end at 0 watts (as defined in the Definitions section above), the sum of all associated powers of any valid episode must be zero. This is a physical property of any cycle of any appliance (Shao, Marwah, and Ramakrishnan 2013; Haasz and Madani 2014). Any candidate episode with a nonzero power sum represents an incomplete cycle for a single appliance or contains events from more than one appliance and, as such, is not valid. For example, the power sequence (+250, +300, -550) is a valid episode since it sums to zero, whereas (+400, -250) is not valid since it has a nonzero sum.

In practice, there is noise in the power values associated with detected events, meaning they will never sum to exactly zero. As such, we mask this noise through a power threshold ν to obtain the formal definition of ϕ_1 in the equation below. We discuss the impact of ν in the Algorithm section below.

$$Given:\ P = (p_{n_1}, p_{n_2}, ..., p_{n_L}) \mid p_{n_i} \in \mathbb{R}$$

$$\phi_1(P) = \begin{cases} True, & \textbf{if } |p_{n_1} + p_{n_2} + ... + p_{n_L}| \leq \nu \\ False, & \text{otherwise} \end{cases}$$

Axiom 2 (ϕ_2): Positive Prefix Sum. The second axiom required for a valid episode is that any sum of a prefix of its power changes must be positive. This captures the property that an appliance's power draw can never be negative, since appliances consume power, they never produce it.

$$\phi_2(P) = \begin{cases} True, & \textbf{if } (\sum_{j=1}^{n_i} p_j) > 0 \quad \forall i \in [1, L-1] \\ False, & \text{otherwise} \end{cases}$$

Axiom 3 (ϕ_3): Minimum Event Power. A third axiom, *minimum event power*, was also introduced by (Shao, Marwah, and Ramakrishnan 2013), which forced any power change in a valid episode to be 10% of its overall power.

We omit ϕ_3 as a constraint since it inhibited episode discovery for three reasons. First, as the length of an episode increases, the power change associated with each of its events represents a diminishing percentage the episode's total power changes. Second, ϕ_3 is not a physical property required for appliances. Finally, this constraint actually prevents the discovery of some episodes, since some high power appliances have small power changes between states.

Iterative Episode Discovery Algorithm

We now detail iterative episode discovery in Algorithm 1. It begins with each event's associated power changes $p_1, p_2, ..., p_n$, max episode length (L_{max}), max horizon window (W_{max}), and a power threshold (ν) for axiom ϕ_1.

The algorithm first initializes empty lists for candidate episodes ($\mathbb{E}_{cand}$) and valid episodes ($\mathbb{E}_{valid}$) (Line 1) and sets the length of episodes to the minimum number possible, $L = 2$ (Line 2). An episode of length $L = 2$ would indicate a Type I appliance turning ON and OFF.

Algorithm 1: Iterative Episode Discovery

Input: Event Power Changes $p_1, p_2, ..., p_n$,
Max Episode Sequence Length L_{max},
Max Window W_{max}, Power Threshold ν
Output: Valid Episodes $\mathbb{E}_{valid}$

1 $\mathbb{E}_{cand}, \mathbb{E}_{valid}$ = {}
2 $L = 2$
3 **while** $L \leq L_{max}$ **do**
4 $W = L$
5 **while** $W \leq W_{max}$ **do**
6 **for** *i = 1 to* $n - W + 1$ **do**
7 $\mathbb{E}_{cand}$ = Candidate episodes of length L in the interval $[p_i, p_{i+W-1}]$
8 **foreach** $P = (p_{n_1}, ..., p_{n_L}) \in \mathbb{E}_{cand}$ **do**
9 **if** $(\phi_1(P)\ AND\ \phi_2(P))$ **then**
10 Add $(p_{n_1}, p_{n_2}, ..., p_{n_L})$ into $\mathbb{E}_{valid}$
11 Mark $p_{n_1}, p_{n_2}, ..., p_{n_L}$ as used
12 $\mathbb{E}_{cand}$ = {}
13 $W = W + 1$
14 $L = L + 1$
15 Return $\mathbb{E}_{valid}$

Next, the algorithm sets the horizon window, W, to the episode length L (Line 4). Using each possible starting event p_i, it generates unexplored episodes of length L in the interval $[p_i, p_{i+W-1}]$ and saves them in $\mathbb{E}_{cand}$ (Lines 6-7).

Each episode in $\mathbb{E}_{cand}$ is tested for validity (Line 9) using axioms ϕ_1 and ϕ_2, described earlier. If valid, the episode is added to $\mathbb{E}_{valid}$ (Line 10), and its events are marked as used so they are not included in any other episode (Line 11).

The horizon window W is progressively increased (Line 13), allowing for an increasing number of external events. Once W_{max} is reached, the sequence length L is then incremented (Line 14) to search for episodes of 1 longer event sequence length, and W is reset to $W = L$ (Line 4). The algorithm is complete once L exceeds L_{max}, and it returns the list of discovered episodes $\mathbb{E}_{valid}$ (Line 15).

During episode validation, it is possible for multiple episodes containing the same event to be valid, meaning different valid episodes could be discovered depending on the order they are tested. We observed this to be rare in our experiments and visited the episodes in the same order (sorted by starting event(s)) to ensure deterministic computation.

Iterative discovery has $O(n^2)$ complexity, the same as brute force, since any candidate episode is generated at most once. In practice, however, iterative discovery is significantly faster since it progressively removes events, rapidly shrinking the search space as the example below illustrates.

Example of Iterative Discovery

Consider the detected events $e_1, ..., e_{19}$ in Figure 1 with $L_{max} = 5$ and $W_{max} = 5$. Iterative discovery starts searching for episodes of length $L = 2$ using a window size of $W = 2$. This generates 18 candidate episodes: $\{(e_1, e_2)$, (e_2, e_3), ..., $(e_{18}, e_{19})\}$. Iterative discovery tests each candidate using ϕ_1 and ϕ_2 and finds 4 valid episodes: $\{(e_3, e_4)$, (e_6, e_7), (e_{12}, e_{13}), $(e_{16}, e_{17})\}$. It marks these 8 events so they are not included in future candidates, leaving 11 events.

Iterative discovery next increments W to $W = 3$. Since events e_2 and e_8 are only separated by a single event now (event e_5), it discovers the valid episode (e_2, e_8). Similarly, when $W = 4$, it finds (e_1, e_9). There are no more events of length $L = 2$, and iterative discovery stops searching for events of length $L = 2$ when $W = 5$ (i.e. $W = W_{max}$). At this point, 6 valid episodes have been discovered, 12 of the 19 events have been marked as used, and 7 events remain for exploration of episodes of length $L = 3$ or more.

Iterative discovery will next search for episodes of length $L = 3$, progressively incrementing W. It finds no valid episodes of length $L = 3$ since there are none in this example. Similarly, it won't find any valid episodes of length $L = 4$, but once it reaches $L = 5$, the algorithm will discover the episode (e_{10},e_{11},e_{14},e_{18},e_{19}).

We omit the number of episodes generated for brevity, but it can be shown that iterative discovery generates only 79 candidates to find all valid episodes of length $L \leq 5$. In contrast, brute force search generates over 16,000 candidates.

Parameter Settings

We experimented with multiple parameter settings for Algorithm 1. For the maximum episode length, L_{max}, we found diminishing returns for $L_{max} > 5$, since episodes of length 6 or more are rare even for high sampling rates. The max horizon window, W_{max}, is challenging to optimize. Events do not occur at regular intervals, meaning a fixed W_{max} corresponds to different time intervals at different points in the event sequence. We experimented in the range $W_{max} \in [5, 30]$, and set $W_{max} = 20$ in our results below. For the power threshold, we used a grid search to explore $\nu \in [5, 100]$ watts in increments of 5 watts. The optimal ν varied for each house between 5 and 25 watts.

Appliance Reconstruction

Previously introduced unsupervised methods cluster detected events (Gonçalves et al. 2011) or steady-state power levels (Shao, Marwah, and Ramakrishnan 2013) prior to episode discovery. In contrast, iterative discovery does no clustering prior to episode discovery and instead associates discovered episodes together to form individual appliances.

For two episodes to be equivalent, they must first have the same length. Second, the average pairwise relative difference between each power change must be less than a specified threshold. All equivalent episodes are associated with the same appliance. We found the best results with a relative difference threshold of 1%. Complex appliances can have varying cycle lengths or different power changes for the same cycle, limiting the accuracy of this approach, but these are challenges for any method. Note that an unsupervised method will not know the underlying label of any appliance (i.e. 'dishwasher') it discovers unless it receives such labels from the user. For our evaluations below, we associated each discovered appliance with the closest labeled appliance.

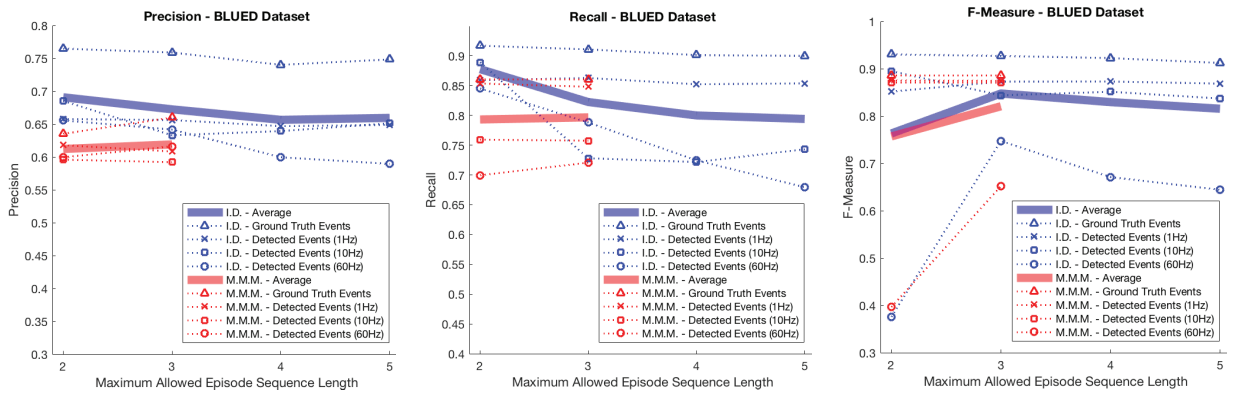

Figure 2: Precision, recall, and F-measure for iterative discovery (I.D. - in blue) and modified motif mining (M.M.M. - in red) using the BLUED dataset. Dotted lines show performance using ground truth events and Bayesian change detected events at 1 Hz, 10 Hz, and 60 Hz, while the solid lines show the average over all 4 sets of detected events. Due to its brute force search, modified motif mining did not return results for episodes of length 4 or 5 within the allotted time frame.

Experimental Setup

Publicly Available Datasets

The performance of any unsupervised disaggregation method will vary depending on the number of appliances in the aggregate data stream, each appliance's complexity, and the sampling rate used to measure real power. As such, we use two publicly available, heavily cited datasets containing 147 appliances in 7 houses for empirical evaluation.

BLUED: The Building-Level fUlly-labeled dataset for Electricity Disaggregation (BLUED) contains power measurements sampled at 60 Hz over a period of 1 week for a single house with 43 appliances (Anderson et al. 2012a). BLUED is unique since it has gone through manual post-processing to establish ground truth event data. In addition to ground truth, we also use events detected by Bayesian change detection (Adams and MacKay 2007; Valovage and Gini 2017) at sampling rates of 1 Hz, 10 Hz, and 60 Hz to measure the impact of sampling rates on episode discovery.

REDD: The Reference Energy Disaggregation Dataset (REDD) contains real power measurements for six houses (Kolter and Johnson 2011). REDD lacks ground truth events and instead records power for each appliance/circuit.

REDD is a more challenging dataset than BLUED. First, power is recorded at lower sampling rate (0.25 Hz). Second, some appliances were recorded over the same circuit, such as *Kitchen_outlets*, *Outdoor_outlets*, and *Outlets_unknown*. With these circuits, it's possible for a method to correctly separate two signals from different appliances but produce lower accuracy since they were recorded on the same meter.

Event Detection

Event detection is required to segment the aggregate power data stream prior to episode discovery. Previously introduced event detection methods include clustering methods such as Dirichlet Process Gaussian Mixture Models (Shao, Marwah, and Ramakrishnan 2013) to change detection methods such as modified Greatest Likelihood Ratio (Anderson et al. 2012b). We use Bayesian change detection (Adams and MacKay 2007) since it is robust to noise and does not require parameter tuning (Valovage and Gini 2017).

Comparison Method

We compare iterative appliance discovery to *temporal motif mining* (TMM) (Shao, Marwah, and Ramakrishnan 2013). While other methods are limited to discovering appliances in isolation (Kolter and Jaakkola 2012) or are only designed to find Type I appliances (Gonçalves et al. 2011), TMM has the ability to identify overlapping episodes of any appliance without pre-existing models, similar to iterative discovery. TMM generates candidate episodes up to length 3 by brute force and tests for validity using the axioms listed above. For fairest comparison with identical assumptions for both methods, we implemented modifications to TMM and refer to the resulting method as *modified motif mining* (MMM).

MMM's modifications are as follows. First, we replaced TMM's event detection approach of Dirichlet Process Gaussian Mixture Models (DPGMM) with Bayesian change detection to detect events and genetic k-means to cluster them, since we observed numerous false positives from DPGMM (Valovage and Gini 2017). Second, we omitted axiom ϕ_3 for the reasons listed earlier. Third, we do not assume the number of appliances is known *a priori*, as this assumption is unrealistic. Finally, we did not apply median filter smoothing as (Shao, Marwah, and Ramakrishnan 2013) used different window sizes for different appliances, an unrealistic assumption when appliances are not known *a priori*.

Note the MMM results below are not directly comparable to TMM, particularly since (Shao, Marwah, and Ramakrishnan 2013) only ran their results on house H1 from the REDD dataset and built synthetic aggregate data streams from randomly generated subsets of only 14 of the 18 appliances, making it impossible to directly recreate their results.

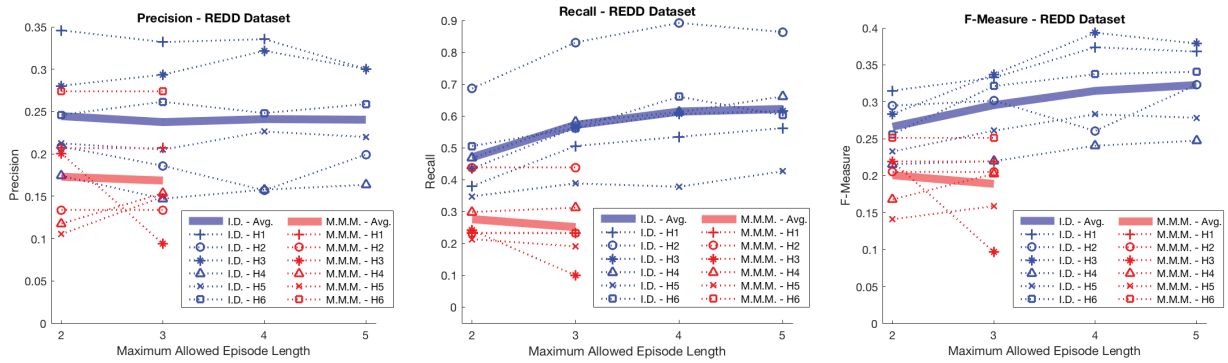

Figure 3: Performance of iterative discovery (I.D. - in blue) and modified motif mining (M.M.M. - in red) using the REDD dataset. Dotted lines show performance on houses H1 through H6 in the dataset, while the solid lines show the average of each method across all houses. Modified motif mining did not complete for episode lengths larger than 3 in the allotted time.

Allotted Runtime

For each house, we allowed each method to run for 24 hours on a conventional laptop. Modified motif mining did not finish for episode lengths longer than 3, while iterative discovery completed in under 5 hours for lengths up to 5.

Performance Metrics

We use two metrics to compare the performance of iterative discovery and modified motif mining. The first metric is F-Measure using disaggregation for each time step. We use the approach to measure true positives (Ψ_{TPi}), false positives (Ψ_{FPi}), and false negatives (Ψ_{FNi}) as a portion of power at each time step as originally proposed by (Shao, Marwah, and Ramakrishnan 2013) and use their same parameter settings of $\theta = 30$ and $\rho = 0.2$. For appliance A_i, F-Measure is defined through precision and recall in the equations below.

$$Prec.(A_i) = \frac{\Psi_{TPi}}{\Psi_{TPi} + \Psi_{FPi}} \qquad Recall(A_i) = \frac{\Psi_{TPi}}{\Psi_{TPi} + \Psi_{FNi}}$$

$$F\_Measure(A_i) = \frac{2}{\frac{1}{Precision(A_i)} + \frac{1}{Recall(A_i)}}$$

The second metric, *total power consumed*, compares the estimated power used by each appliance with the total actual power used. Total power consumed can mask errors, but is useful to report to consumers (Kolter and Johnson 2011).

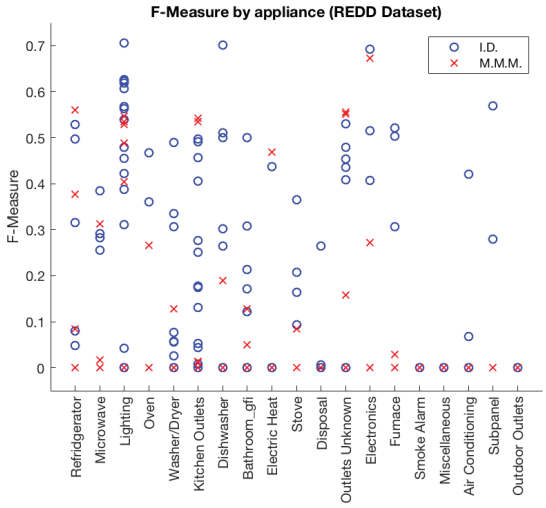

Figure 4: REDD F-Measure by appliance. In general, iterative discovery found more appliances than motif mining.

Results

Results on the BLUED Dataset

Figure 2 displays the results on BLUED obtained by iterative discovery and modified motif mining using the ground truth detected events and Bayesian change detection events at 1 Hz, 10 Hz, and 60 Hz. Motif mining was only able to complete searches for episodes of length 2 and 3 within the allotted time due to its brute force search. This is unsurprising, as the number of candidate episodes was nearly 10^{14} for some of the event sequences. In contrast, iterative discovery found episodes up to length 5 well within the allotted time.

Both methods performed best when using BLUED's ground truth events. This is expected, since this is the best possible input data for any episode discovery method, as there are minimal errors in the ground truth events.

Using events detected by Bayesian change detection, performance was highest for both methods at the 1 Hz sampling rate and lowest at 60 Hz. This occurs because higher sampling rates produce more detected events with larger variations. This produces false positives from candidate episodes that happen to fit axioms ϕ_1 and ϕ_2, but actually contain events from different appliances. False positives can also create more false negatives, since an actual episode containing an event that has been used can't be discovered.

Iterative discovery experienced a decrease in performance when searching for episodes of lengths 4 and 5. In BLUED's ground truth events, there are no episodes of length longer

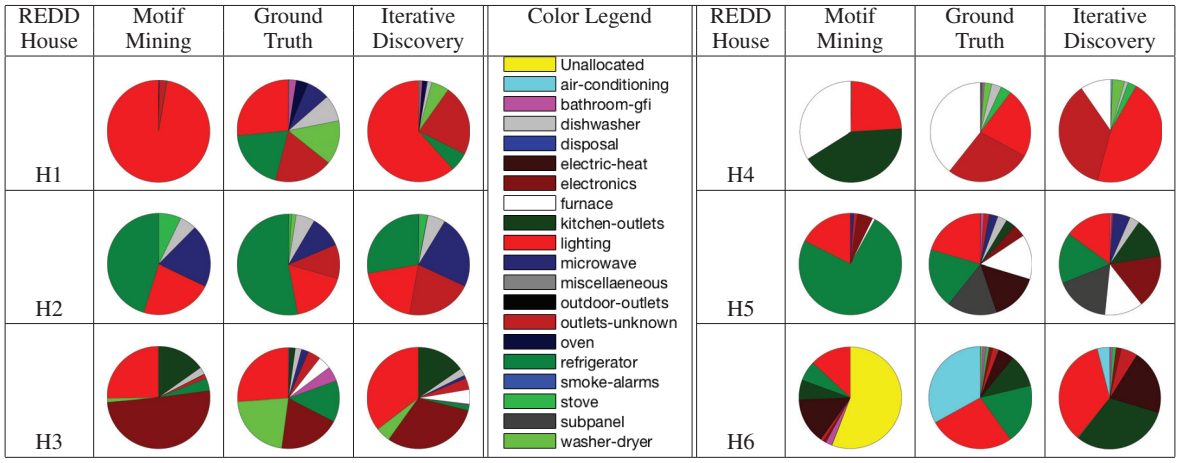

Figure 5: Pie charts showing the estimated and actual energy consumed for each house in REDD. Ground Truth columns show the actual energy consumed, while the Iterative Discovery and Motif Mining columns show the estimated energy usage using each method. Estimations from both methods contained roughly the same error for houses H2 and H3. Iterative discovery's estimations are more accurate for houses H1, H4, H5, and H6, where it correctly identified more episodes and appliances.

than 3, and even for higher sampling rates, episodes of these lengths are rare compared to false positives generated. Overall, iterative discovery found longer episodes and performed as well as or better than modified motif mining on BLUED using less computation time.

Results on the REDD Dataset

Figure 3 shows precision, recall, and F-measure for the 6 houses in the REDD dataset along with the average performance across all houses. Performance for both methods was significantly lower than on BLUED due to the challenges the REDD dataset poses to unsupervised disaggregation described above. Low precision was caused by multiple false positives, since both methods incorrectly combined some events from different appliances into episodes. Recall was better, particularly on house H2 which is expected, as house H2 has 9 appliances, the fewest of any house in REDD.

Iterative discovery performed better than modified motif mining with episodes of length 2 and 3. When the search was expanded to episodes of length 4 and 5, this increased the F-measure of iterative discovery on average, while motif mining was unable to complete its search in the allotted time.

We note that the original temporal motif mining was only run on house H1 in REDD. Modified motif mining implemented here on house H1 achieved precision, recall, and F-measure around 0.2, similar to the performance reported in (Shao, Marwah, and Ramakrishnan 2013).

Figure 4 breaks down performance of each method by appliance type for the entire REDD dataset. Iterative discovery identified continuously variable Type III appliances, such as furnaces, dishwashers, appliances plugged into kitchen outlets, washers, and dryers with greater F-measure than motif mining. These are the most difficult to discover since their power draw varies over an infinite number of states (Zoha et al. 2012). Iterative discovery also found appliances motif mining could not, including the subpanel in H5, air conditioners in H4 and H6, and disposals in H2, H3, and H5.

Finally, Figure 5 shows actual power consumed by appliance for each house and estimates produced by iterative discovery and modified motif mining. The error in these estimations varies but is roughly the same for both methods for houses H2 and H3, while iterative discovery's estimations for houses H1, H4, H5, and H6 tend to be more accurate.

In house H1, motif mining attributes the vast majority of used power to the 3 light sets in the house. This happens for 2 reasons. First, motif mining discovers a disproportionate number of episodes for the lights in H1 compared to the other appliances. Second, these discovered episodes include numerous false positives. This also occurs in H4 and H5 to a lesser extent. Iterative discovery suffers less from this and is also able to correctly find more episodes from appliances like the stove, washer-dryer, and dishwasher in H4 and the subpanel in H5. In H6, motif mining had unallocated power due to fewer false positives relative to the number of episodes discovered, resulting a significant portion of power that could not be attributed to any single appliance.

In summary, Figures 3-5 show iterative discovery produced better results than motif mining with less computation time. While accuracy of the total power estimates from iterative discovery is limited, we are not aware of better results obtained through learning that is completely unsupervised with no prior models or assumptions.

Conclusions and Future Work

This paper has introduced iterative appliance discovery, the first appliance discovery technique that can identify appliances overlapping in operation in an unsupervised manner with no appliance models. Identifying the simplest appliances first allows for better accuracy, faster computation, and scalability to unprecedented sequence lengths.

Iterative discovery could be expanded to power sources, such as solar panels and batteries like the Telsa Powerwall,

by modifying axiom 2 to allow for negative prefix sums produced when such systems generate power. For widespread deployment, further work is also needed to automatically tune parameters and enable effortless setup for consumers.

References

Adams, R. P., and MacKay, D. J. 2007. Bayesian online changepoint detection. *Technical report, University of Cambridge, Cambridge, UK* arXiv:0710.3742.

Anderson, K.; Ocneanu, A.; Benitez, D.; Carlson, D.; Rowe, A.; and Berges, M. 2012a. BLUED: A fully labeled public dataset for event-based non-intrusive load monitoring research. In *Proceedings of the 2nd KDD workshop on data mining applications in sustainability (SustKDD)*, 1–5.

Anderson, K. D.; Bergés, M. E.; Ocneanu, A.; Benitez, D.; and Moura, J. M. 2012b. Event detection for non intrusive load monitoring. In *IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society*, 3312–3317. IEEE.

Carrie Armel, K.; Gupta, A.; Shrimali, G.; and Albert, A. 2012. Is disaggregation the holy grail of energy efficiency? the case of electricity. *Energy Policy*.

Frankel, D.; Heck, S.; and Tai, H. 2013. Sizing the potential of behavioral energy-efficient initiatives in the U.S. residential market. McKinsey & Company.

Gonçalves, H.; Ocneanu, A.; Bergés, M.; and Fan, R. 2011. Unsupervised disaggregation of appliances using aggregated consumption data. In *The 1st KDD Workshop on Data Mining Applications in Sustainability (SustKDD)*.

Granade, H. C.; Creyts, J.; Derkach, A.; Farese, P.; Nyquist, S.; and Ostrowski, K. 2009. Unlocking energy efficiency in the U.S. economy. McKinsey & Company.

Gupta, S.; Reynolds, M. S.; and Patel, S. N. 2010. Electrisense: single-point sensing using EMI for electrical event detection and classification in the home. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, 139–148. ACM.

Haasz, V., and Madani, K. 2014. *Advanced Data Acquisition and Intelligent Data Processing*. River Publishers.

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

Kolter, J. Z., and Jaakkola, T. 2012. Approximate inference in additive factorial hmms with application to energy disaggregation. In *International Conference on Artificial Intelligence and Statistics*, 1472–1482.

Kolter, J. Z., and Johnson, M. J. 2011. REDD: A public data set for energy disaggregation research. In *Workshop on Data Mining Applications in Sustainability (SIGKDD), San Diego, CA*, volume 25, 59–62. Citeseer.

Kolter, J. Z.; Batra, S.; and Ng, A. Y. 2010. Energy disaggregation via discriminative sparse coding. In *Advances in Neural Information Processing Systems*, 1153–1161.

Lobaccaro, G.; Carlucci, S.; and Löfström, E. 2016. A review of systems and technologies for smart homes and smart grids. *Energies* 9(5):348.

Malik, F. H., and Lehtonen, M. 2016. A review: Agents in smart grids. *Electric Power Systems Research* 131:71–79.

Mannila, H.; Toivonen, H.; and Verkamo, A. I. 1995. Discovering frequent episodes in sequences extended abstract. In *1st Conference on Knowledge Discovery and Data Mining*.

Mannila, H.; Toivonen, H.; and Verkamo, A. I. 1997. Discovery of frequent episodes in event sequences. *Data mining and knowledge discovery* 1(3):259–289.

Parson, O.; Ghosh, S.; Weal, M. J.; and Rogers, A. 2012. Non-intrusive load monitoring using prior models of general appliance types. In *AAAi*.

Shaloudegi, K.; György, A.; Szepesvári, C.; and Xu, W. 2016. SDP relaxation with randomized rounding for energy disaggregation. In *Advances in Neural Information Processing Systems*, 4978–4986.

Shao, H.; Marwah, M.; and Ramakrishnan, N. 2013. A temporal motif mining approach to unsupervised energy disaggregation: Applications to residential and commercial buildings. In *Proc. AAAI Conference on Artificial Intelligence (AAAI)*.

Tomkins, S.; Pujara, J.; and Getoor, L. 2017. Disambiguating energy disaggregation: A collective probabilistic approach. *International Joint Conference on Artificial Intelligence*.

Valovage, M., and Gini, M. 2017. Label correction and event detection for electricity disaggregation. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, 990–998. International Foundation for Autonomous Agents and Multiagent Systems.

Zoha, A.; Gluhak, A.; Imran, M. A.; and Rajasegarar, S. 2012. Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey. *Sensors* 12(12):16838–16866.