A Simple Model-Driven Approach to Energy Disaggregation

Guoming Tang*, Kui Wu*, Jingsheng Lei†, and Jiuyang Tang‡
*Dept. of Computer Science, University of Victoria, B.C., Canada
†School of Computer and Information Engineering, Shanghai University of Electric Power, Shanghai, China
‡Sci. & Technol. on Information Systems Engineering Lab, National University of Defense Technology, Changsha, China

***Abstract*—Energy disaggregation is to discover the energy consumption of individual appliances from their aggregated energy values. To solve the problem, most existing approaches rely on either appliances' signatures or their state transition patterns, both hard to obtain in practice. Aiming at developing a simple, universal model that works without depending on sophisticated machine learning techniques or auxiliary equipments, we make use of easily accessible knowledge of appliances and the sparsity of the switching events to design a Sparse Switching Event Recovering (SSER) method. By minimizing the total variation (TV) of the (sparse) event matrix, SSER can effectively recover the individual energy consumption values from the aggregated ones. To speed up the process, a Parallel Local Optimization Algorithm (PLOA) is proposed to solve the problem in active epochs of appliance activities in parallel. Using real-world trace data, we compare the performance of our method with that of the state-of-the-art solutions, including the popular Least Square Estimation (LSE) methods and a recently-developed machine learning method using iterative Hidden Markov Model (HMM). The results show that our approach has an overall better performance in both detection accuracy and overhead.**

I. Introduction

Energy disaggregation, also known as non-intrusive appliance load monitoring (NIALM), aims to learn the energy consumption of individual appliances from their aggregated energy consumption values, *e.g.*, the total energy consumption of a house. With accurate energy disaggregation, the house owner can 1) learn how much energy each appliance consumes, 2) take necessary actions to save energy, and 3) participate in utility demand response programs. Furthermore, with smart meters broadly deployed in many countries, sufficiently high resolution of energy data can be collected, making it feasible to develop efficient energy disaggregation solutions.

Due to its critical meaning, the energy disaggregation problem has attracted more and more attention since 1980s. Recently, it has also drawn attention from both large electronics companies and small start-ups, such as Intel, Belkin, GetEmme, and Navetas. While many methods have been developed for energy disaggregation, according to [15], no solutions work well for all types of household appliances. They either work poorly for new types of appliances or require complex machine learning method to learn appliances' (latent) features.

A. Related Work

Tremendous research efforts have been devoted to solving the energy disaggregation problem. The existing approaches can be roughly divided into two categories: *signature based* methods and *state transition based* methods.

Most approaches are based on appliances' signatures, *i.e.*, specific features such as the real/reactive power, current, and voltage of running appliances [5]. These methods need the support of high sampling rate and build either steady or transient signal features of appliances with labeled training datasets. The signal features are treated as the appliances' signatures [10], [8], based on which event detection schemes are developed to detect appliances' on/off as well as different running states. The detected events are ascribed to certain appliances' activities via classification [2], [13]. In addition to time-domain signal features, spectral analysis has also been adopted to search for appliances' signatures in the frequency domain [9], [4]. Nevertheless, the signatures are hard to obtain without particular machine learning techniques or auxiliary measurements.

Some other methods made use of state transition in appliances' activities. The Hidden Markov Model (HMM) has been adopted to model the state transition patterns of appliances. The hidden states of each appliance at each time instant are predicted by inference algorithms, such as the Viterbi algorithm, with the observed emission probabilities [12]. Nonnegative sparse coding has been proposed to solve the energy disaggregation problem in [6], in which a training process is needed to obtain the basis vector related to the state transition patterns of different appliances. These methods usually need a large number of trainings, and thus are time consuming. In addition, the performance highly relies on the pattern of appliances' activities in the training datasets, and as such the performance may vary significantly from test to test.

B. Our Contributions

The related work shows that most current energy disaggregation methods lack practicality for common house occupants. Therefore, aiming at establishing an easy-to-use, universal model for energy disaggregation, we make the following contributions in this paper.

- Instead of relying on appliances' *signature*, we use the information of *rated power*, which is easy to obtain, *e.g.*, from the user's guide of appliances. With experimental evaluation, we show that the method is robust even if this information is not very accurate.

- Based on the simple power model and the sparsity property of appliance activities, we establish a universal Sparse Switching Event Recovering (SSER) optimization model, and tries to minimize the total variation of on/off switching events, which has *never* been explored before to solve the energy disaggregation problem.
- We develop a Parallel Local Optimization Algorithm (PLOA) to solve SSER, which can significantly reduce the computational complexity of the original problem and is guaranteed to obtain the optimal solution if some weak hypotheses hold.
- We build a small-scale energy monitoring platform for a group of household appliances, and evaluate our method using the real-world trace data collected over the platform. The results indicate that our approach has an overall better performance than state-of-the-art solutions.

II. System Model

A. Power Pattern

We focus on the aggregated power readings of a number of appliances in a house, and arrange them from time $t = 1$ to T as an *aggregated power vector*[1],

$$X := [X_1, X_2, \cdots, X_T]^T . \tag{1}$$

The power pattern of an appliance indicates the energy consumption value when it is turned on or in stand-by state. In this paper, we use a simple power model which can be easily obtained from the user's guide or the specification of an appliance. We represent the power pattern of an appliance n by a tuple (I_n, P_n, Θ_n), where I_n is its stand-by power, P_n is its rated power, and Θ_n is its power deviation.

Assume that a house is equipped with N appliances. We define a *stand-by power vector* to represent their stand-by powers as

$$I := [I_1, I_2, \cdots, I_N]^T , \tag{2}$$

a *rated power vector* to represent their rated powers as

$$P := [P_1, P_2, \cdots, P_N]^T , \tag{3}$$

and a *power deviation vector* to represent their power deviations as

$$\Theta := [\Theta_1, \Theta_2, \cdots, \Theta_N]^T . \tag{4}$$

Definition 1. *Given a house with a certain number of appliances, we call the sum of the appliances' stand-by power, denoted by P_0, as the* ***baseline power*** *of the house, i.e., $P_0 = \|I\|_1$.*

Note that virtually all appliances' stand-by power could be found from users' manual, technical specification or public websites such as [3]. Theoretically, P_0 should be constant, which is the minimum power of the house at any time instant. In practice, however, there are small variations in P_0 due to inaccurate stand-by power specification, thus it is possible that the actual power could be below the baseline power.

At time instant t, given the state vector of all appliances S_t, the aggregated power reading $X_t, (t = 1, 2, \ldots, T)$, is bounded by:

$$\begin{aligned} (\mathbf{1} - S_t)^T I + S_t^T (P - \Theta) \le X_t, \\ (\mathbf{1} - S_t)^T I + S_t^T (P + \Theta) \ge X_t, \end{aligned} \tag{5}$$

where $\mathbf{1}$ is the all-one vector. In other words, the following constraints hold:

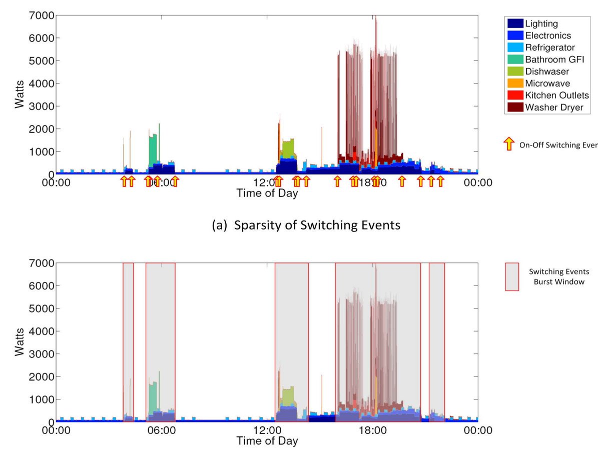
$$\begin{aligned} X - S^T (P + \Theta) - (\mathbb{I} - S)^T I \le \mathbf{0}, \\ S^T (P - \Theta) + (\mathbb{I} - S)^T I - X \le \mathbf{0}, \end{aligned} \tag{6}$$

where $\mathbb{I}$ is the N-by-T all-one matrix.

B. Sparsity of Switching Events

Fig. 1[2] shows an example of energy consumption and appliances on/off switching events in a typical house during one day. From the figure, we can see that:

- As shown in Fig. 1-a, the appliances do not switch on/off frequently in the whole time period.
- Most switching events happened in a small number of time intervals, which we call *active epochs* (refer to Section III-A for formal definition) and are illustrated with shaded windows in Fig. 1-b.



(a) Sparsity of Switching Events

(b) Active Epochs of Switching Events

Fig. 1. Energy consumption and appliances' on/off switching events in a house over the course of a day [7]

We denote the on/off states of N appliances from time $t = 1$ to T with a *state matrix*, S, defined as

$$S := \begin{bmatrix} S_1^{(1)} & S_2^{(1)} & \cdots & S_T^{(1)} \\ S_1^{(2)} & S_2^{(2)} & \cdots & S_T^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ S_1^{(N)} & S_2^{(N)} & \cdots & S_T^{(N)} \end{bmatrix}, \tag{7}$$

in which $S_t^{(n)}$ represents the on/off state of the n-th appliance at time t, and $S_t^{(n)} \in \{0, 1\}$ with $S_t^{(n)} = 1$ indicates the n-th appliance is on and 0 otherwise.

[1]Without loss of generality, all vectors in the paper are column vectors. When T is used as the superscript of a vector/matrix, it means the transpose of the vector/matrix in this paper.

[2]The figure is from [7] with slight modification for better illustration.

Then, the on/off switching events of the N appliances from $t = 2$ to T can be indicated by an *event matrix*, ΔS, calculated as

$$\Delta S = SD, \tag{8}$$

where D is a differential matrix with size of N-by-$(N-1)$:

$$D := \begin{bmatrix} -1 & & & & \\ 1 & -1 & & & \\ & 1 & \ddots & & \\ & & \ddots & -1 & \\ & & & 1 & -1 \\ & & & & 1 \end{bmatrix} \tag{9}$$

The element of event matrix $\Delta S_t^{(n)} \in \{-1, 0, 1\}$, with $\Delta S_t^{(n)} = 1$ or -1 indicating a switching on or off event of the n-th appliance at time t, respectively, and 0 no switching event. Since the sampling rate of current smart meters nowadays is relatively high, we neglect the situation where an appliance has a series of switching events within a sampling interval, *i.e.*, $|\Delta S_t^{(n)}| < 2$.

Assertion 1. *According to our real-world observations, ΔS is a **sparse** matrix.*

C. Sparse Switching Events Recovering

To compute the energy consumption of an individual appliance during a time period, we also need to know its activities (states changing) along the timeline. Based on the power model of appliances and sparsity feature of switching events, we formulate the following problem to recover the on/off states of individual appliance at each time instant.

- **Input:** Aggregated power vector X, power pattern (I, P, Θ).
- **Output:** State matrix S, *i.e.*, the on/off states of all appliances along the timeline.

A **Sparse Switching Event Recovering (SSER)** model is established to recover the states of N appliances from time $t = 1$ to T.

$$\begin{aligned} \min_{S} \quad & \mathbf{TV}(\Delta S) \\ s.t. \quad & X - S^T(P + \Theta) - (\mathbb{I} - S)^T I \leq \mathbf{0}, \\ & S^T(P - \Theta) + (\mathbb{I} - S)^T I - X \leq \mathbf{0}, \end{aligned} \tag{10}$$

where ΔS is defined by (8) and $\mathbf{TV}(\cdot)$ denotes the *total variation* of the event matrix calculated by

$$\mathbf{TV}(\Delta S) := \sum_{n=1}^{N} \sum_{t=1}^{T} \left| \Delta S_t^{(n)} \right|. \tag{11}$$

After obtaining the on/off states of each appliance along the timeline, we can estimate its power readings with its rated power at each time instant. Therefore, we can get an approximate estimation of the power consumption of each appliance. This is equivalent to solving the original energy disaggregation problem. Note that all appliances contributing to the aggregated power vector need to take into consideration in SSER model. Otherwise, the accuracy of recovered states will decrease. This may be a limitation when applying our approach, as some appliances may be unknown or forgotten.

The total variation (TV) minimization is a classical approach to recovering a sparse matrix. It has been widely applied in signal restoration, image denoising, and compressive sensing [11]. To the best of our knowledge, however, it has not been explored in the context of energy disaggregation. Unlike other optimization methods, such as least square fitting [13], total variation minimization is a type of least absolute deviations fitting, which has been proved to be more robust in various applications.

III. Parallel Local Optimization

There were significant research efforts to solve the total variation minimization problem. Nevertheless, the form of total variation in our case is a discrete version and involves integer variables, which is much harder. We prove that solving SSER is NP-hard (refer to [14] for details), so it is hard to find the optimal solution, especially when the time interval is large. Nevertheless, the active epochs of on/off events suggest that we can perform optimization in a smaller, local time window.

A. Detection of Active Epochs

Definition 2. *An **active epoch** of a house is defined as a time interval from the time when the aggregated power of the house jumps above the baseline power until the time when the aggregated power drops below the baseline power.*

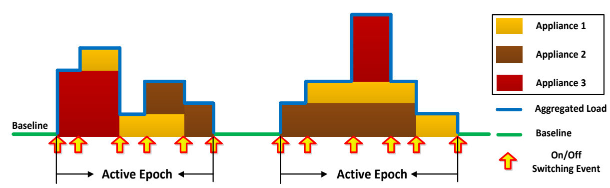


Fig. 3. A sketch map to illustrate the concepts of active epoch and baseline power using three appliances

Fig. 3 is a sketch map of switching activities and power readings of three appliances with constant power, in which the concepts of baseline power and active epoch are illustrated. Algorithm 1 shows the pseudo code of detecting active epochs.

B. Parallel Local Optimization Algorithm (PLOA)

Without loss of generality, we take aggregated load data of N appliances from time $t = 1$ to T as an example to show the major steps of PLOA.

Step 1: Detect all active epochs along the timeline with Algorithm 1. Denote the set of active epochs as $W = \{W_1, W_2, \cdots, W_k\}$.

Step 2: In the active epoch starting at t with the length of ℓ, solve the following optimization problem to obtain $S_{t:t+\ell}$.

$$\begin{aligned} & \min_{S_{t:t+\ell}} \ \mathbf{TV}(S_{t:t+\ell} D_{t:t+\ell}) \\ & s.t. \\ & X_{t:t+\ell} - (S_{t:t+\ell})^T(P + \Theta) - (\mathbb{I}_{t:t+\ell} - S_{t:t+\ell})^T I \leq \mathbf{0}, \\ & (S_{t:t+\ell})^T(P - \Theta) + (\mathbb{I}_{t:t+\ell} - S_{t:t+\ell})^T I - X_{t:t+\ell} \leq \mathbf{0}, \end{aligned} \tag{12}$$

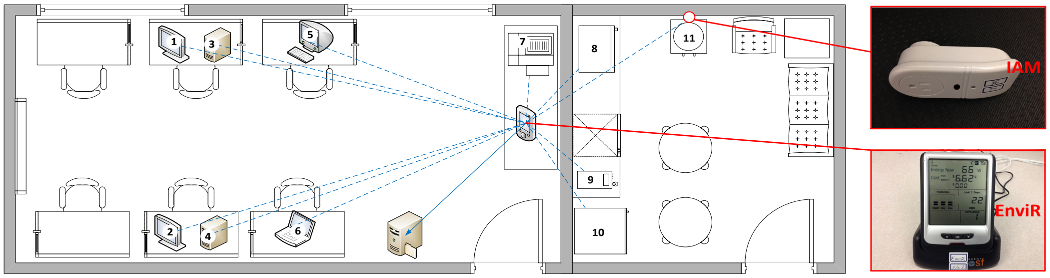


Fig. 2. Energy monitoring platform, monitored appliances, and measuring devices

Algorithm 1 Active Epoch Detection

Input: Aggregated power vector X, baseline power P_0.
Output: Set of Active epochs, W.

$t = 1, k = 0$
while $t \leq T$ **do**
  $start = end = t$
  while $X_{end} > P_0$ and $end < T$ **do**
    $end = end + 1$
  end while
  if $end > start$ **then**
    $k = k + 1$
    $W_k = [start, end]$
  end if
  $t = end + 1$
end while
return $W = \{W_1, W_2, \cdots, W_k\}$

where $S_{t:t+\ell}$ is a N-by-ℓ submatrix of S, $D_{t:t+\ell}$ is a ℓ-by-$(N-1)$ submatrix of D, $\mathbb{I}_{t:t+\ell}$ is a N-by-ℓ submatrix of $\mathbb{I}$, and $X_{t:t+\ell}$ is a vector containing the aggregated power readings of all appliances from time t to time $t+\ell$.

Step 3: Perform Step 2 on the k active epochs to obtain a group of k solutions in parallel. Since outside of active epochs, appliances are considered as stand-by, a complete state matrix $S_{1:T}$ can thus be built.

We can show that the computational complexity to solve (12) is $O(2^{N \cdot \ell})$. Since $\ell \ll T$ as shown in Fig 1, the (mixed-integer linear program) problem can be solved efficiently, using tools such as *CVX* 2.0 with a *Gurobi* engine [1].

Theorem 1. *Assume that the global optimal solution to SSER in (10) is S^*, and the solution obtained from POLA is $\hat{S}$, if both solutions are unique, then $\hat{S} = S^*$.*

Proof: For an arbitrary active epoch starting at t with the length of ℓ, assume that $\hat{S}_{t:t+\ell}$ is the unique local optimal solution obtained via (12). Assume that the global optimal solution to SSER in (10) is S^*, and the sub-matrix constructed by the t-th to $(t+\ell)$-th columns of S^* is $S^*_{t:t+\ell}$. We prove the theorem by contradiction.

Assume that

$$\hat{S}_{t:t+\ell} \neq S^*_{t:t+\ell}. \tag{13}$$

Then, the following inequality must hold

$$S^*_{t:t+\ell} D_{t:t+\ell} \geq \hat{S}_{t:t+\ell} D_{t:t+\ell}. \tag{14}$$

Therefore, there must exist another global solution S^{**}, in which the j-th column is

$$S^{**}_j = \begin{cases} \hat{S}_j \ , j \in [t, t+\ell], \\ S^*_j \ , j \notin [t, t+\ell], \end{cases} \tag{15}$$

such that

$$S^{**} D \leq S^* D. \tag{16}$$

Obviously, (16) is contradictory to the assumption that S^* is the *uniquely* global optimal solution to SSER. Therefore, the assumption (13) is not true. As a result, we have

$$\hat{S}_{t:t+\ell} = S^*_{t:t+\ell}. \tag{17}$$

Outside the active epochs, PLOA treats all appliances as stand-by, the TV value is 0 in $\hat{S}$. Since the TV value cannot be negative, the TV value obtained with PLOA is the minimum and must be the same as that obtained with the global optional solution.

Overall, if the global optimal solution is unique, for any time instant t, no matter whether t is in an active epoch or outside active epochs, the state vector $\hat{S}_t \in \hat{S}$ must be equal to the state vector $S^*_t \in S^*$, which means $\hat{S} = S^*$. ■

Given T aggregated power readings generated by N appliances that can be broken into k active epochs with maximum size w, the computational complexity of the original SSER problem (10) is $O(2^{N \cdot T})$. With PLOA, solving the local optimization problem in (12) k times results in the time complexity upper bounded by $O(k \cdot 2^{N \cdot w})$. Considering that the number of appliances N is constant and $w \ll T$, PLOA significantly cuts down the computational complexity.

IV. Experimental Evaluation

We use the real-world trace data collected from our energy monitoring platform to evaluate our method, and compare it with 1) a signature based approach, the Least Square Estimation based integer programming method [13] and 2) a state transition based approach, the iterative Hidden Markov Model [12].

A. Data Collection

Currently, there are some research based datasets available for energy disaggregation. Most of them, however, 1) are circuit-oriented rather than appliance-oriented, such as the REDD dataset [7], or 2) lack power information of appliances. To avoid these problems, we setup an energy monitoring platform where we can gather information according to our demand.

We monitored the appliances' energy consumption in a typical laboratory and a lounge room in the fifth floor of Engineering/Computer Science building at the University of Victoria (UVic). Using an off-the-shelf solution developed by Current Cost (*http://www.currentcost.com*), we recorded the real-time power of laptops, desktops and some household appliances. Each appliance's real power was measured every 6 seconds by the device called Individual Appliance Monitor (IAM), and the measurement results were transmitted via wireless to a display server (EnviR), which can display and temporarily store the collected data. Then, the data in EnviR were sent to our data server. The platform, monitored appliances, and measuring devices are shown in Fig. 2.

The power consumption information of appliances are summarized in Table. I[3], where the rated and stand-by powers are learned from the users' manual or according to [3], and the power deviations are estimated from the collected power data of each appliance. One may be concerned that the estimation of power deviation in practice is inaccurate. With experimental study, however, we will show that our method is resilient to inaccurate power deviation estimations in Section IV-C.

TABLE I
POWERS INFORMATION OF APPLIANCES

ID	Appliance	Mode	Rated Power (*Watts*)	Power Deviation (*Watts*)	Stand-by Power (*Watts*)
1	LCD-Dell	1	25	5	0
2	LCD-LG	1	22	5	0
3	Desktop	1 2	40 50	15 20	3
4	Server	1	130	20	10
5	iMac	1 2	35 50	5 10	3
6	Laptop	1 2 3	15 30 70	5 10 10	1
7	Printer	1 2 3	400 700 900	50 80 100	2
8	Microwave	1 2 3	1000 1200 1700	100 100 100	2
9	Coffee Maker	1 2 3	700 900 1100	100 100 100	2
10	Refrigerator	1 2	115 350	15 10	5
11	Water Cooler	1 2 3	65 380 450	5 10 10	3

B. Performance Evaluation

We collected data for three months from the energy monitoring platform, and one-week data were used for performance evaluation. The evaluation metrics are defined as:

- *Energy Disaggregation Accuracy (EDA)*: It indicates the accuracy of assigning correct power values to corresponding appliances and was also used in [7].

$$EDA := 1 - \frac{\sum_{n=1}^{N} \left\| X^{(n)} - \hat{S}^{(n)} P_n \right\|_1}{\left\| X \right\|_1}, \tag{18}$$

  where $X^{(n)}$, $\hat{S}^{(n)}$ and P_n represent the true energy consumption vector, the estimated state vector, and the rated power of the n-th appliance, respectively, and X is the aggregated power vector.
- *State Prediction Accuracy (SPA)*: It indicates the accuracy of estimating the states of appliances.

$$SPA := 1 - \frac{\sum_{n=1}^{N} \left\| S^{(n)} - \hat{S}^{(n)} \right\|_1}{N \cdot T}, \tag{19}$$

  where $S^{(n)}, \hat{S}^{(n)}$ represent the true state vector and the estimated state vector of the n-th appliance, respectively, and N, T represent the number of appliances and the number of samples, respectively.
- Running time (*R.T.*) and memory usage (*RAM*)[4]: They indicate the overhead on running time and memory space, respectively.

Since the performance of the iterative HMM method depends on model training, we run this method multiple times over different sizes (w.r.t. number of samples) of training datasets (denoted as *training size*). The average performance is calculated over all the runs, and the best and the worst performance are the best and the worst outcomes among all the runs, respectively.

With the same prior knowledge in Table I, the performance of the three methods are summarized in Table II. In addition, as illustrated in Fig. 4, we also look into the *overall energy disaggregation accuracy* of the three methods, which indicates the energy contribution of each appliance to the total energy consumption in the whole time period. From the results, we can draw the following conclusions:

- In term of accuracy, our SSER method performs much better than the LSE based method and slightly better than the iterative HMM method in average.
- In term of overhead, our SSER method and the LSE method are at a comparative level for running time and system memory usage. While the memory usage of the iterative HMM method is similar to that of the other two method, its running time is much higher.
- The performance of the iterative HMM method is subject to the training process and may have a large variation in accuracy and in running time.

C. Robustness Test

In practice, the rated power of an appliance can be easily learned. However, we may not precisely estimate the power deviation of an appliance working under a certain mode. As

[3]Considering some appliances may have multiple operating modes (rated powers), we regard each as a *virtual appliance*, such that an individual appliance with multiple modes was split into multiple virtual ones in the SSER model.

[4]We implemented the three methods with *Matlab* 8.0 and run them with 32-bit Windows OS with $3.4GHz$ CPU and $4GB$ RAM.

TABLE II
ACCURACY AND OVERHEAD OF ENERGY DISAGGREGATION, USING SPARSE SWITCHING EVENT RECOVERING (SSER), LEAST SQUARE ESTIMATION (LSE) BASED INTEGER PROGRAMMING AND ITERATIVE HIDDEN MARKOV MODEL (HMM)

Methods \ Metrics	ACCURACY		OVERHEAD		
	EDA	*SPA*	*Training Size*	*R.T.(second)*	*RAM(MB)*
SSER	61.12%	69.62%	–	865.4	596.8
LSE	33.40%	45.67%	–	619.3	581.9
HMM (average)	55.27%	67.47%	2116	3721.3	558.6
HMM (best)	67.26%	71.29%	600	1299.7	557.9
HMM (worst)	41.09%	61.27%	3200	7089.6	561.4

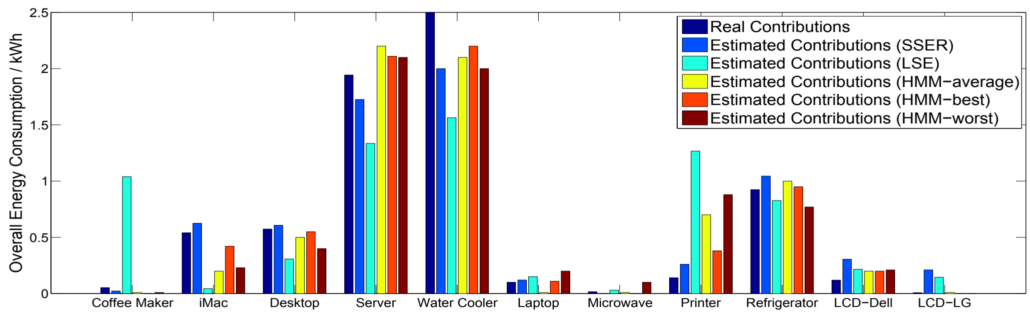


Fig. 4. Actual and estimated energy contributions of each appliance to the total energy consumption for one-week time period.

such, we test the performance of our method, assuming that the power deviations of appliances are inaccurate. We replace Θ with $\rho \cdot \Theta$ in the SSER model, so that the estimated power deviations can be changed by regulating ρ.

TABLE III
ACCURACY OF ENERGY DISAGGREGATION USING SSER WITH INACCURATE ESTIMATION ON POWER DEVIATION

ρ \ Metrics	*EDA*	*SPA*
0.8	55.28%	70.37%
0.9	60.33%	70.27%
1.0	61.12%	69.62%
1.1	56.94%	71.15%
1.2	59.59%	72.24%

We changed the value of ρ from 0.8 to 1.2, causing a parametric error of power deviation up to 20%. Part of the outcomes are shown in Table III. We can see that the accuracy does not change too much when the parameter error varies, indicating that our method is robust to parameter estimation.

V. CONCLUSIONS

In this paper, a simple, universal model for energy disaggregation was proposed. By making use of readily available information of appliances, we built a sparse switching event recovering model based on the sparsity of appliances' switching events. Furthermore, we used the active epochs of switching events to develop a parallel local optimization algorithm to solve our model efficiently. In addition to analyzing the complexity and correctness of our algorithm, we tested our method with the real-world trace data from an energy monitoring platform. The test results demonstrated that our method can achieve better performance than the state-of-the-art solutions, including the Least Square Estimation (LSE) method and the machine learning method using iterative Hidden Markov Model (HMM).

REFERENCES

[1] CVX. Matlab software for disciplined convex programming. http://cvxr.com/cvx/, 2013. [Online; accessed in 23-July-2013].

[2] M. Dong, P. C. Meira, W. Xu, and W. Freitas. An event window based load monitoring technique for smart meters. *Smart Grid, IEEE Transactions on*, 3(2):787–796, 2012.

[3] EPA. A tool of home product finder by energy star. http://www.energystar.gov/productfinder/, 2013. [Online; accessed in 18-December-2013].

[4] S. Gupta, M. S. Reynolds, and S. N. Patel. 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*, pages 139–148. ACM, 2010.

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

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

[7] J. Z. Kolter and M. J. Johnson. Redd: A public data set for energy disaggregation research. In *proceedings of the SustKDD workshop on Data Mining Applications in Sustainability*, pages 1–6, 2011.

[8] C. Laughman, K. Lee, R. Cox, S. Shaw, S. Leeb, L. Norford, and P. Armstrong. Power signature analysis. *Power and Energy Magazine, IEEE*, 1(2):56–63, 2003.

[9] S. B. Leeb, S. R. Shaw, and J. L. Kirtley Jr. Transient event detection in spectral envelope estimates for nonintrusive load monitoring. *Power Delivery, IEEE Transactions on*, 10(3):1200–1210, 1995.

[10] L. K. Norford and S. B. Leeb. Non-intrusive electrical load monitoring in commercial buildings based on steady-state and transient load-detection algorithms. *Energy and Buildings*, 24(1):51–64, 1996.

[11] S. Osher, A. Solé, and L. Vese. Image decomposition and restoration using total variation minimization and the h 1. *Multiscale Modeling and Simulation*, 1(3):349–370, 2003.

[12] O. Parson, S. Ghosh, M. Weal, and A. Rogers. Non-intrusive load monitoring using prior models of general appliance types. In *AAAI*, 2012.

[13] K. Suzuki, S. Inagaki, T. Suzuki, H. Nakamura, and K. Ito. Nonintrusive appliance load monitoring based on integer programming. In *SICE Annual Conference, 2008*, pages 2742–2747. IEEE, 2008.

[14] G. Tang, K. Wu, J. Lei, and J. Tang. Plug and play! a simple, universal model for energy disaggregation. http://arxiv.org/abs/1404.1884, 2014.

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