Estimating Building Consumption Breakdowns using ON/OFF State Sensing and Incremental Sub-Meter Deployment

Deokwoo Jung and Andreas Savvides
Department of Electrical Engineering
Yale University
New Haven, CT 06511
{deokwoo.jung, andreas.savvides}@yale.edu

Abstract

This paper considers the problem of estimating the power breakdowns for the main appliances inside a building using a small number of power meters and the knowledge of the ON/OFF states of individual appliances. First we solve the breakdown estimation problem within a tree configuration using a single power meter and the knowledge of ON/OFF states and use the solution to derive an estimation quality metric. Using this metric, we then propose an algorithm for optimally placing additional power meters to increase the estimation certainty for individual appliances to the required level. The proposed solution is evaluated using real measurements, numerical simulations and by constructing a scaled down proof-of-concept prototype using binary sensors.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms,Design, Experimentation, Measurement, Performance, Theory

Keywords

Energy breakdowns, Electric Load Estimation, Electricity Consumption Monitoring

1 Introduction

Understanding the energy and power usage breakdowns is a fundamental component for next generation intelligent buildings. Besides providing feedback to inhabitants on how to conserve and identifying inefficiencies, detailed and close to real-time consumption accountability is expected to empower buildings with better understanding and control of their loads. In addition to identifying and controlling waste, fine grained load control would enable buildings to provide regulation service reserves (or demand/response services) back to the grid. The provision these services implies the commitment of the building to stand by and provide a service by reducing or increasing its consumption up to a pre-agreed amount according to a set of commands transmitted to the building by the grid Independent Service Operator (ISO). For the buildings, this generates some revenue that amounts to reduced net energy costs. For the grid, the provision of regulation services on the building side helps to make it easier to leverage intermittent availability of renewable energy from wind solar and other more intermittent sources.

A recent survey we conducted in newer buildings at our institution has shown that already installed state-of-the-art building automation solutions such as those provided by Johnson Controls, Honeywell and Siemens already provide sophisticated interfaces for managing loads. The total aggregate consumption of each building is readily available from their local meters through standard protocols, mainly modbus. The main information missing for regulation service reserves and waste reduction is a more detailed understanding of load properties and consumption breakdowns.

Motivated by this opportunity, in this paper we provide a methodology for estimating consumption breakdowns with lightweight appliance state sensing. In particular, we focus on estimating the breakdown of dynamic consumption components inside a building using only knowledge of the total consumption and the ON/OFF binary state of energy consumers (appliances) inside the building. ON/OFF state information can either be sensed with simple dry-contact sensors, more intelligent wireless sensor nodes that infer the state of an appliance from indirect sensing as done by Kim et. al. in [10] or direct human inputs to the user interface of our system. The proposed approach is complementary to a variety of solutions ranging from processing transient load signatures [8] to approaches that use power meters at the sockets [13] and indirect sensing [10] as described in more detail in section 3. Its goal is to provide a scalable solution with very lightweight sensing that can also complemented by human inputs. Our solution is general and can be applied to residential settings where humans can manually enter state information into the system, all the way to large buildings where a large network of sensors would provide a low-cost retrofit solution for monitoring power consumption.

The main challenge in solving this problem is twofold. First, the power consumption profiles for the individual appliances are not always constant. For example, a certain

SenSys'10, November 3–5, 2010, Zurich, Switzerland.

Figure 1. **Active RFID dry contact sensor from RF code**

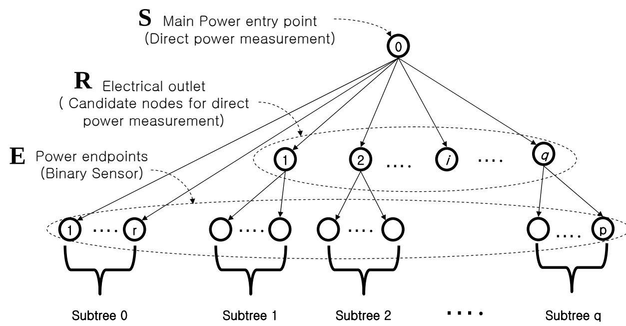


Figure 2. **Tree representation of binary sensors and a power meter in electricity network in building**

light-bulb may have a constant power consumption of $60W$ whereas a washing machine or a refrigerator may have a fluctuating consumption during the time it is on. Second, we cannot actively control the ON/OFF state of appliances to condition the dataset and this results in a lot of ambiguity that our algorithm has to handle.

Our main contribution is the development of an algorithm that periodically estimates the required breakdowns by carefully considering the variations in loads and the diversity in the binary set of appliance states. At each cycle the algorithm selects between two estimates computed from two alternative datasets and selects the one with the highest confidence based on a data quality metric. This results in a lightweight setup that does not require the exhaustive deployment of power meters on all appliances. To achieve the required level of certainty in larger deployments, we also provide an incremental algorithm for deploying additional meters where needed. This algorithm essentially decomposes a binary data set into subsets according to the topology of electrical outlets and strives to maximize diversity of the appliance state dataset by incrementally deploying additional meters. The proposed algorithms are validated with real datasets collected from our testbeds.

Our presentation is organized as follows. Section 2 describes the system setup we consider. Section 3 surveys the related work and section 4 formulates our problem. Section 5 describes our solution for a single meter deployment. Section 6 provides an algorithm for deploying additional meters. Section 7 presents our evaluation results and section 8 concludes the paper and outlines our future work.

2 System Setup and Model

To estimate consumption breakdowns per unit time, we assume that the total consumption is known at the central meter. In our experimental setup this is measured at the mains fusebox with a TED 5000 Series home energy monitor from “The Energy Detective” that provides a stream of measurements in XML format at a rate of 1Hz. Appliances are instrumented with Active RFID dry contact sensors from RF Code similar to the one shown in Figure 1. These sensors detect the devices’ ON/OFF states changes as events and propagate them wirelessly to a central computer where the readings from the energy monitor together with the recorded states are processed to estimate breakdowns.

To solve the problem, we model the hierarchical structure of a building power network as a three-tier tree structure where the power consumption at a parent node is equal to the sum of the power consumptions of its children as shown in Figure 2. More elaborate tree structures such as the ones in [9] can also be supported. Throughout this paper our discussion uses a three tier tree for simplicity of exposition.

As shown in Figure 2, the mains power meter is at the root of the tree, all the outlets are in the middle tier and the appliances attached to the outlets are found on the leaf nodes of the tree. We first solve for the case where only a single power meter is available in the system and then we use the metric derived from this solution to propose an algorithm for incrementally deploying additional power meters at the middle tier of the tree so as to achieve the best tradeoff between estimation quality and the number of power meters. Following this scheme, the subtrees with the most uncertainty are collapsed into a single node with high certainty by inserting a power meter at that outlet.

3 Related Work

Existing approaches for electricity load monitoring can be broadly classified into two main categories, direct and indirect sub-metering. Direct sub-metering measures power consumption at each electric load end-point using a power meter. Various direct energy monitors are commercially available such as the *WATTS UP.Net* plug and play power meter [6] and integrated sub-metering solutions from Tendril *Tendril* [5], and the *The Energy Detective* [2]. Several Zigbee based sub-metering solutions are also surfacing from companies like *Millenial Net* [3], *Sensicast* [4], and *Arched Rock* sub-metering solutions [1]. Jiang et. al. have developed a AC power meter platform, *ACme node* [9] and deployed 38 *ACme node*s for several months in a building to extract temporal, spatial, and personal energy usage patterns [9]. Lifton et. al. have designed *Plug*, an intelligent power strip to collect a wide range of measurements including current and voltage [13]. They demonstrated that such measurements can be used to infer appliance ON/OFF states. Although the direct-metering system can provide precise consumption breakdowns, its deployment a large electricity networks comes at considerable cost and effort.

The popular alternative to direct-metering is to apply indirect-metering approaches; placing a single or few power meters at hot-spots such as the electricity entry point in the building or a room and inferring energy usage of individual loads with more sophisticated signal processing algorithms. Significant work has been done in such system so far. Hart et. al. has developed the Non-Intrusive Load Monitoring (NILM) system which only monitors the total electricity load and disaggregates it by detecting specific load signatures of individual appliances [8]. In addition to this early work, various signal processing techniques have been explored to identify the power state of specific appliances more accurately. The *transient event detector* has been proposed to identify ON/OFF events of a specific appliance types by analyzing the transient events of electrical loads [12, 14]. It decomposes electrical signals into spectral domains and uses pattern recognition techniques to match templates of patterns. Patel et. al. have proposed algorithm of classifying the electrical noise on residential power lines to detect the ON/OFF state change events of a particular appliance [15].

Although the above indirect-metering approaches have been shown to perform well in small residential homes, they would have limitations in larger electrical networks where large numbers of appliances have very similar transient loads. This is mainly because it might not be feasible to separate transient events among the same type of appliances. Moreover, transient event detection requires an exhaustive database of signal templates and that would render it infeasible for some settings. Besides the fact that such a proposition is costly, the adoption of internal power control schemes modern electric appliances makes transient event detection even more challenging. To address the problem, Kim et. al have developed *ViridiScope* system [10]. This system uses indirect sensors in addition to conventional power meters to monitor power consumption. The authors detect ambient signals emitted from appliances and infer their power consumption using a set of intelligent autonomous calibration techniques.

Our work aims to achieve a good compromise between deployment costs and the computed energy breakdown accuracies by replacing power meters with binary sensors. Increasing the ratio of binary sensors to power meters in the system essentially shifts the energy monitoring system from the direct-metering to indirect-metering system. Our work is similar to [10] in that we also build up on the ideas of indirect sensing. The fundamental difference however is that our problem is a data disaggregation problem, not a calibration problem as in [10]. The primary goal is to estimate the energy consumption breakdown of appliances rather than reconstructing their (fine-grained) power consumption profiles. Another important difference is that we put our emphasis on developing a scalable and computationally efficient estimation algorithm, something that is treated as a black box in [10]. The advantage of using binary sensors is that they are not required to send a raw sensor measurement data to a data center like the *ViridiScope* system. Instead, binary sensors are much cheaper and only need to do thresholding to detect ON/OFF transitions. By keeping the input information simple, it also becomes possible to use our algorithm without any sensor deployment. Instead, users can just enter the ON/OFF information on a portal and contact their own experiments in their buildings to determine the average power consumptions of their appliances. Furthermore, in larger buildings the ON/OFF events can be collected directly from the building automation system, making our solution functional as a retrofit solution that does not require additional equipment deployment.

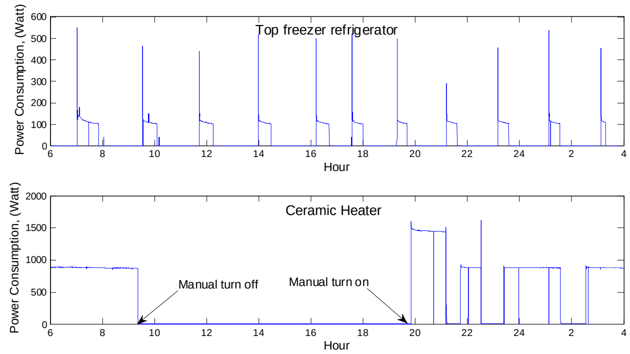


Figure 3. Power consumption profile of top freezer refrigerator (top) and ceramic heater (bottom)

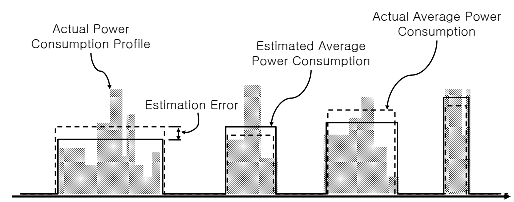


Figure 4. The illustrative example of the estimated and the actual average power consumption of an appliance

4 Sensing Model Formulation

4.1 Assumptions

To solve this problem we assume that the sampling rate, T_s is smaller than estimation Period, T_{est} and that the ON/OFF states of all appliances are detected without error, that is, a lower-level alrgotithm or subsystem is responsible for making the correct detection. We also assume that appliances have a constant vampire power when they are off and that the active power consumption of an appliance is much greater than its vampire power. Finally, we also assume that the variance is less than the average of active power consumption of an appliance and ON/OFF sequence of appliances are weakly correlated.

4.2 Problem Overview

Estimating the energy usage breakdown is equivalent to estimating average power consumption per appliance over a given observation time. Therefore, in our problem instead of making direct consumption measurements and then averaging, we directly compute the average with the knowledge of the aggregate consumption and appliance state. We attempt to accurately estimate average consumption over short time intervals keeping in mind that in many appliances the consumption during their ON state is not uniform. As an example, Figure 3 shows typical power consumption profiles for a top freezer refrigerator and a temperature controlled ceramic heater. In the Figure, the refrigerator shows a consistent periodic power consumption pattern drive by the ON/OFF states of its compressor. The temperature controlled heater on the other hand has a less consistent power profile mainly because the ON/OFF events are initiated by both a human and a controller. It is turned off manually around 9am and back on again around 8pm. The heater initally consumes more power to reach the target temperature and then less power to maintain it.

The above examples suggest that in the absence of explicit sub-metering, the breakdown estimation algorithm should adaptively compute the average power consumption over the duration of each ON event for each device as in the illustrative example of Figure 4. Although the consumption fluctuates over the ON state, our algorithm will aim to compute the average power consumption (shown with a solid line in the Figure) over the ON interval. We refer to this as the non-stationary property of active power profiles. The goal of the algorithm is to minimize the estimation error which is defined as the difference between the estimated and the actual average power consumption for the ON duration.

This estimation problem is challenging because of the *the collinearity of binary data* and *the non-stationary property of active power profiles*. In an ideal case scenario where only one appliance is ON at a given time, the binary data set would have *zero collinearity* and the estimates will be very accurate. If however, all appliances are switched ON at the same time, the worst case scenario, it is much harder to discern the average power consumption for individual appliances from the binary (ON/OFF) data set.

The above extreme cases point to an important tradeoff that our algorithm aims to exploit. If the device power profiles are stationary, we can estimate more accurately if we collect more data by extending the estimation period. If however devices have many power modes (i.e less stationary profiles), then the accuracy would be higher if the estimation period is smaller and considers less samples. In other words, we can always estimate an average power consumption if we observe the power profile long enough. However, if it is non-stationary, the observation time might not improve estimation accuracy. Using the data observed during the estimation period often generates better estimates than using the whole observed data set.

Intuitively, in the ideal cases where a single appliance is on at any given time and all appliances have stationary loads breakdowns can be accurately computed using only binary knowledge of ON/OFF states and a single power meter at the building's main fusebox. In reality however, electrical appliances have a degree of collinearity, and non-stationary power profiles. These rapidly compromise performance as more devices enter the system. Our solution tries to mitigate this issue by dynamically selecting an appropriate time window to consume the state measurements based on the diversity of the observed dataset. This is done by evaluating an estimation quality metric.

For the cases where the required accuracy is not reached, we provide an additional algorithm that uses our metric to make decisions where additional power meters need to be placed. As described lated on in section 6, these decisions are made incrementally in a conservative manner placing one meter at a time until the required accuracy is achieved with the minimum number of additional meters.

Figure 5 depicts an illustrative example of the placement

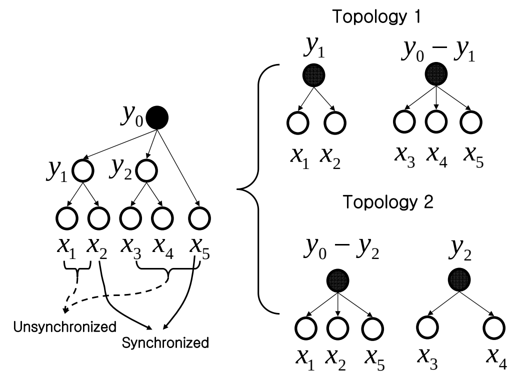


Figure 5. Illustrative example of power meter topology

problem. On the left, the system has a main power entry point, two electrical outlets and five appliances organized as a tree topology. The dark node denotes the presence of a power meter at that node. In this layout, if the required certainty is not reached, we would have to add an additional power meter at one of the outlets $y1$ or $y2$. The two alternative topologies are shown on the right side of Figure 5. Although both topologies have the same number of subtrees and leaf nodes, their data quality can be different. For example, if we assume that ON/OFF sequence of appliances in each set of $\{x_1, x_2\}$ and $\{x_3, x_4, x_5\}$ are perfectly unsynchronized to each other while perfectly synchronized for $\{x_2, x_5\}$. Then we can estimate the average power consumption of each appliance without error for topology 1 since the state appliances in each subtree change one at a time. In topology 2 however, it is much more challenging to computer the average power consumption of x_2 and x_5 since their ON/OFF state changes are completely overlapped.

By solving the above problem of incremental deployment using the binary state data observations and our quality metric, we essentially avoid the brute force approach of exhaustively having to measure the power consumption at each outlet to evaluate estimation performance. Instead, the estimation performance at each subtree is predicted from the observed binary sensor data. This takes place by first computing a data quality metric using a single meter at the root of the tree and later using the metric for guiding the deployment of additional meters on the tree. This process essentially collapses low-certainty subtrees into single high-certainty nodes through the addition of a power meter at the root of the subtree.

4.3 Problem Statement

Our breakdown estimation problem can be stated as follows: *Given a time-stamped data set of binary power states for all appliances and their cumulative power consumption measured at the fusebox, estimate the average energy consumption of each appliance over a given time interval.*

Mathematically, we assume that the number of appliances is p and for each i_{th} appliance we have ON/OFF binary data, $x_i(t) \in \{0, 1\}$ and its actual power consumption, $P_i(t) \in \mathbf{R}^+$ at time t where $i \in \{1, \cdots, p\}$. The p-tuple for the ON/OFF states of leaf nodes at time t by the column vector is denoted by $\mathbf{x}(t) = [x_i(t) \cdots x_p(t)]^T$. Furthermore, we assume that appliances with strongly correlated ON/OFF states are considered as a single appliance (e.g a desktop computer and its LCD monitor are considered as the computer).

Ideally, our objective would be to estimate average power consumption of individual appliances over a certain time duration. This can be formulated as the optimization problem in (1), which minimizes the sum of the mean square errors of the difference between the measured $\bar{P}_i[k]$ and estimated $\hat{\bar{P}}_i[k]$ active power consumption of each appliance over a given interval k.

$$\min \sum_{i=1}^{p} MSE(\hat{\bar{P}}_i[k]) = E\left[\sum_{i=1}^{p}(\bar{P}_i[k]-\hat{\bar{P}}_i[k])^2\right] \quad (1)$$

The above problem however is not solvable because we don't have actual consumption measurements at each node ith node, that is, $\bar{P}_i[k]$ is not known. Before suggesting an alternative formulation we first introduce some notation details.

We would like to generate a new estimate at a pre-set periodic interval T_{est}, using a collection of samples from the power meter that take place an interval T_s where $T_s << T_{est}$. The collection of appliance binary states is collected in an event-driven manner and it is assumed to be much slower than the total power consumption sampling interval at the meter. The former is in the order of tens of seconds to tens of minutes while the latter takes place once a second. A new energy breakdown estimate is computed at the end of each period, $\{T_{est}, 2T_{est}, \cdots, kT_{est}, \cdots\}$, which index by k as $\mathbf{t}_k = \{(k-1)T_{est}+T_s, (k-1)T_{est}+2T_s, \cdots, kT_{est}\}$. Let $\bar{P}_i[k]$ denote the average active power consumption of $P_i(t)$ during the kth estimation period, which is denote in (2)

$$\bar{P}_i[k] = \frac{\sum_{t\in\mathbf{t}_k} P_i(t)x_i(t)}{\sum_{t\in\mathbf{t}_k} x_i(t)} \quad (2)$$

We define the active power profile of ith appliance is stationary if it converges the constant value, $\bar{P}_i$ over periods, i.e $\lim_{k\to\infty}\bar{P}_i[k] = \bar{P}_i$. Otherwise, it is non-stationary.

Using the above, we can formulate our problem in terms of the total consumption at the root of the tree as

$$\min MSE(y(t)) = E\left[\frac{1}{T}\sum_{t=t_0}^{t_0+T}(y(t)-\hat{y}(t))^2\right] \quad (3)$$

where $y(t)$ is the measured consumption total at the root at time t and $\hat{y}(t)$ is the sum of the estimated consumption for each leaf node in the tree. This problem can now be solved by expressing $\hat{y}(t)$ in terms of the individual power consumptions.

The instantaneous power consumption of ith appliance, $P_i(t)$ using $x_i(t)$ and $\bar{P}_i[k]$ as shown in (4)

$$P_i(t) = (\bar{P}_i[k]+\varepsilon_{i1}(t))x_i(t)+\varepsilon_{i0}(t)(1-x_i(t)) \quad (4)$$

where $\varepsilon_{i1}(t)$ is an error between the observed active power consumption and the expected value, and $\varepsilon_{i0}(t)$ is the vampire load consumed by the appliance while it is switched off.

t	$x_1(t)$	$x_2(t)$	$x_3(t)$	$y(t)$
1	0	0	1	62
2	0	0	1	60
3	1	0	0	120
4	0	1	1	380
5	1	0	1	160
6	0	1	1	371
7	1	1	1	469
8	0	0	1	56
9	0	1	1	357

Table 1. Illustrative example of a data table with 9 aamples: it shows ith appliance's ON/OFF state $x_i(t)$, the total power consumption in watt $y(t)$ at each time t.

Using $y(t) = \sum_i P_i(t)$ and the model in (4), the total power consumption at kth estimation period can be described as follows

$$y(t) = \sum_{i=1}^{p}(\bar{P}_i[k]-\varepsilon_{i0}(t))x_i(t)+\sum_{i=1}^{p}\varepsilon_{i0}(t)+\sum_{i=1}^{p}\varepsilon_{i1}(t)x_i(t) \quad (5)$$

Equation (5) can be further simplified by assuming that the vampire load of each appliance in the OFF state is constant thus $\varepsilon_{i0}(t) = \varepsilon_{i0}$. We also assume that such loads are negligible compared to the ON power consumption. $\bar{P}_i[k]-\varepsilon_{i0}(t) \approx \bar{P}_i[k]$. After these simplifications, the total power consumption can be described as the simple linear model shown in (6).

$$y(t) = \bar{\mathbf{P}}_{\mathbf{k}}\mathbf{x}(t)+P_0+e(t) \quad (6)$$

where $P_0 = \sum_i \varepsilon_{i0}(t)$ (the total vampire load), $e(t) = \sum_i \varepsilon_{i1}(t)x_i(t)$ (the error of the linear model), and $\bar{\mathbf{P}}_{\mathbf{k}} = (\bar{P}_1[k], \cdots, \bar{P}_p[k])$ (a vector of appliance's average active power consumption at period k).

5 Breakdown Estimation using a Single Power Meter

5.1 Solving the Linear Optimization Problem

We first describe our algorithm with an illustrative example using a simple dataset shown in Table 1(a). The data set is comprised of 9 samples from three binary sensors $x_1(t)$, $x_2(t)$ and $x_3(t)$ and their corresponding total measured power at the meter $y(t)$. The power measurement in the $y(t)$ column is the running average of all meter samples collected while the three binary sensors remain in the same state.

In the example of Table 1 the goal is to estimate the average active power consumptions of the three appliances, $\{\bar{P}_1, \bar{P}_2, \bar{P}_3\}$ over the total time period where the samples were obtained by minimizing the mean square error (MSE) in (3).

For practical purposes, the samples in the table can be represented in a more compact form where the samples with the same state (i.e equivalent binary string in a row) can be compacted in a single row. For instance row 1 in the table can be combined with rows 2 and 8, row 4 can be combined with row 9 and so on. This will result in a sample table that has at most 2^p rows for all p appliances. In this compact form samples are aggregated over the estimation period and sample index t is replaced by k. The number of times a row occurs is also tracked in a counter variable n_k. The variance of the running average in column $y(t)$ for each row sample is also recorded. Applying this more representation to Table 1, would result in a more compact table of 5 rows.

The structure of the sample table reveals two main properties that our algorithm aims to exploit. First, samples that have fewer appliances in the ON state provide more information than samples with multiple appliances on, hence they should carry more weight in the estimation. Second, the variance of the meter readings running average in $y(t)$ contains useful information about the stationarity of loads. This could also be exploited to give heavier consideration to stationary loads. Both cases suggest that our problem should solve a weighted version of equation 3. To estimate breakdowns we therefore first solve a weighted linear optimization problem and provide its solution in matrix form. After that we focus our discussion on the design of an appropriate weight scheme.

Using our samples from Table 1, we can formulate the weighted MSE function for our example as

$$L(\bar{\mathbf{P}}) = \sum_{k=1}^{5} w_k(\bar{y}_k - \mathbf{x}_k(1)P_1 - \mathbf{x}_k(2)P_2 - \mathbf{x}_k(3)P_3)^2 \qquad (7)$$

where $\mathbf{x}_k(i)$ represents the ith bit our state and and w_k represents a new weight coefficient for the entire sample row $\mathbf{x}_k$.

By taking partial derivatives and solving the multivariate equations for L in (7) we can obtain the following matrix equation of $\bar{\mathbf{P}}$, $(\mathbf{X}'\mathbf{W}\mathbf{X})\bar{\mathbf{P}} = \mathbf{X}'\mathbf{W}\overline{\mathbf{Y}}$. For the example in Table 1, each matrix is defined by

$$\mathbf{X} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \mathbf{W} = diag\begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \end{pmatrix}, \overline{\mathbf{Y}} = \begin{pmatrix} 59.3 \\ 369.3 \\ 120 \\ 160 \\ 469 \end{pmatrix}$$

We can easily generalize the solution of the simple example for p binary sensors (appliances) and n samples. Let us assume that a data set collected from p binary sensors and one power meter. If m number of distinctive $\mathbf{x}_k$'s are observed, the corresponding Weighted Mean Table is $\mathbf{TB} = [\mathbf{W}_{m\times 1}|\mathbf{X}_{m\times p}|\overline{\mathbf{Y}}_{m\times 1}]$. The estimate of $\mathbf{P}$ is obtained by solving the following optimization problem with positive constraint as shown in (8).

$$\hat{\mathbf{P}} = \underset{\mathbf{P}\geq 0}{argmin}\ \|(\mathbf{X}'\mathbf{W}\mathbf{X})\mathbf{P} - \mathbf{X}'\mathbf{W}\overline{\mathbf{Y}}\| \qquad (8)$$

where $\mathbf{W}$ is $m \times m$ diagonal matrix whose ith diagonal element will be defined in the next section. Let define $\tilde{\mathbf{P}} = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}\overline{\mathbf{Y}}$. Then the optimal solution of (8) is simply $\hat{\mathbf{P}} = \tilde{\mathbf{P}}$ if $\tilde{\mathbf{P}} \geq 0$.

Now that we have given the general form of the solution in the next two subsections we focus on the design of a suitable weight scheme that exploits state diversity and the observed consumption variance. Furthermore, we also introduce a performance metric that allows us to make decisions on which data window to consider based to account for the non-stationarity of loads.

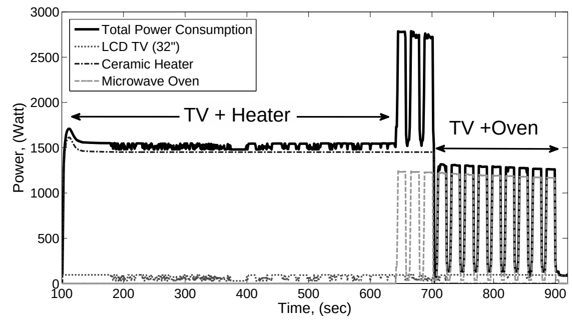


Figure 6. Example of non-uniform variance of active power consumption of appliances: the variance of Microwave Oven is significantly larger than TV's and Heater's, which causes a large estimation error of average power consumption of TV at (TV+Oven) power state

5.2 Designing Weights

Accounting for Diversity An appropriate weight should favor states that have less appliances on, thus less ambiguity, as well as states that are observed more frequently and over longer time intervals. For the former we use the reciprocal of the number of 1s in each row as part of the weight $\frac{1}{\sum_i \mathbf{x}_k(i)}$, and for the latter we take the number of samples obtained for each state over the total samples in the estimation period, thus the weight becomes $w_k = \frac{n_k}{\sum_i \mathbf{x}_k(i)}$.

The above weight coefficient can successfully capture the diversity of binary ON/OFF data, but still does not take account of non-uniformities in active power consumption of individual appliances. In fact, a large dynamic range of active power consumptions often contributes more estimation error than a noisy binary state.

As and example, Figure 6 shows the power profile of an LCD TV, a Microwave Oven, and a Heater as well as their total power consumption. The Microwave Oven shows a large dynamic range of active power consumption (40$\sim$1200 Watt) during its ON state. At two composite power states of (TV+Heater) and (TV+Oven), their weight coefficients are equal given the same number of samples. However, it is much more difficult to estimate the average consumption of the TV when both the TV and the Over are ON than when the TV and Heater are ON because the total power consumption of the TV and Oven together is significantly more noisy than the other case. In order to reflect such an effect on estimation, the weight coefficient has to take account of the variance of active power consumption of each appliance.

Variance Sum Weight Matrix Let α_i^2 denote the variance of active power consumption of ith appliance. The intuitive extension of the coefficient weight, $w_k = \frac{n_k}{\sum_i \mathbf{x}_k(i)}$ is to use a weighted binary sum of the variances, $\sum_i \alpha_i^2 \mathbf{x}_k(i)$ in place of a unit binary sum, $\sum_i \mathbf{x}_k(i)$. The weight coefficient effectively scales the influence of samples on the estimation at each composite binary state according to active power consumption variances. We use a variance sum weight matrix for the estimation where its weight coefficient at a sample, $\mathbf{x}_k$ is defined as follows in (9). We do note however that α_i^2 is not observable in our setup and needs to be estimated as we describe next.

$$w_k = \frac{n_k}{\sum_{i=1}^{p} \alpha_i^2 \mathbf{x}_k(i)} \tag{9}$$

Per Appliance Active Power Consumption Variance Estimation Our algorithm estimates the variances from the weighted mean table using the variance of total power consumption and ON/OFF states of appliances. From the weighted mean table, we can derive the following equation.

$$Var(y_{ki}|\mathbf{x}_k) = \frac{1}{n_k}\left(\sum_{i=1}^{p} \alpha_i^2 \mathbf{x}_k(i) + \sum_{i \neq j}^{p} Cov(\bar{P}_i \mathbf{x}_k(i), \bar{P}_j \mathbf{x}_k(j))\right) \tag{10}$$

where y_{ki} is ith sample of $y(t)$ at $\mathbf{x}_k$ and $Cov(x,y)$ is the covariance of x and y.

The covariance sum, $\sum_{i,j} Cov(\cdot)$ is relatively much smaller than the variance sum, $\sum_i \alpha_i^2 \mathbf{x}_k(i)$ assuming that $\mathbf{x}_k(i)$ and $\mathbf{x}_k(j)$ are approximately independent. Equation (10) implies that we can estimate α_k's by formulating a convex optimization problem with the following two constraint conditions.

- Degree of freedom: the minimum number of samples to compute the variance of $y(t)$ at $\mathbf{x}_k$ is $n_k > \sum_i \mathbf{x}_k(i)$
- Variance bounds: The standard deviation of the active power consumption of an appliance must be greater than 0 and less than its average active power consumption

Assuming that we have the weighted mean table of m samples, i.e $\mathbf{TB} = [\mathbf{W}_{m\times 1}|\mathbf{X}_{m\times p}|\overline{\mathbf{Y}}_{m\times 1}]$ and let M denote a subset of rows, $\{1, \cdots, m\}$ which satisfy the first constraint, $n_k > \sum_i \mathbf{x}_k(i)$. Then the variance of active power consumption can be estimated by solving the following optimization problem for $\alpha_1^2, \cdots, \alpha_p^2$.

$$\min_{\alpha_1^2,\cdots,\alpha_p^2} \sum_{k \in M} \left(\frac{1}{\sum_i \mathbf{x}_k(i)} (n_k Var(y_{ki}) - \sum_{i=1}^{p} \alpha_k^2 \mathbf{x}_k(i)) \right)^2 \tag{11}$$
$$s.to \;\; 0 \le \alpha_i^2 \le \bar{P}_i^2,$$

For computing the upper bound of variance, $\bar{P}_i^2$ we use the estimates for the cumulative data set, from a weighted mean value table that aggregates measurements over the entire measurement interval. Note that the covariance term in (10) increases if more appliances are ON. To take it into consideration, we assign a weight $\frac{1}{\sum_i \mathbf{x}_k(i)}$ in (11).

5.3 Metric Driven Data Selection

The other source of variation we can observe in our data set is related to the stationarity properties of loads. For stationary loads, more data results in better estimates. For non-stationary loads however, this is not always true. To account for this, we compute estimates over two alternative datasets and then apply an estimation quality metric to select estimate with the least error. The two datasets are maintained in two alternative versions of the weighted mean table. The first one keeps the entire data set over the entire collection time, and we call the estimates computed from this table as the cumulative power estimates or $\hat{\bar{\mathbf{P}}}_{cma}$. The second table contains measurements collected over the current estimation period which we denote as $\hat{\bar{\mathbf{P}}}_{cur}$.

k	$w_k = \frac{n_k}{\sum_i \mathbf{x_k}(i)}$	$\mathbf{x_k}(1)$	$\mathbf{x_k}(2)$	$\mathbf{x_k}(3)$	$\bar{y}_k$	n_k
1	1/3	1	1	1	469	1
2	1/2	0	1	1	357	1
3	1/1	0	0	1	56	1

(a)

k	$w_k = \frac{n_k}{\sum_i \mathbf{x_k}(i)}$	$\mathbf{x_k}(1)$	$\mathbf{x_k}(2)$	$\mathbf{x_k}(3)$	$\bar{y}_k$	n_k
1	3/1	0	0	1	59.3	3
2	3/2	0	1	1	369.3	3
3	1/1	1	0	0	120	1
4	1/2	1	0	1	160	1
5	1/3	1	1	1	469	1

(b)

Table 2. Weighted Mean Value Table constructed from, (a) the data set of the 3rd estimation period, $t = \{7, 8, 9\}$, and (b) the entire data set of $t = \{1, \cdots, 9\}$ in Table 1

Adaptive Data Selection Considering the samples in Table 1 assuming the estimation period, $T_{est} = 3T_s$. At the third estimation period, $t = \{7T_s,\ 8T_s,\ 9T_s\}$ we can construct the weighted mean table either from the partial data set observed from the current period as shown in the Table 2(a) or the cumulative data set of $t = \{T_s, \cdots, 9T_s\}$ as shown in Table 2(b). By solving optimization problem given the weighted mean table of cumulative data set, the estimates are $\bar{P}_3 = 57.7$, $\bar{P}_2 = 309.38$, and $\bar{P}_1 = 111.88$, which denoted by $\hat{\bar{\mathbf{P}}}_{cma} = [111.88,\ 309.3,\ 57.7]$. Meanwhile, the binary matrix, $\mathbf{X}$ becomes a triangular matrix in Table 2(a). Therefore, we can simply compute the average power consumption without ambiguity, which is $\bar{P}_3 = 56$, $\bar{P}_2 = 357 - 56 = 301$, and $\bar{P}_1 = 469 - 357 = 112$, that denoted by $\hat{\bar{\mathbf{P}}}_{cur} = [112,\ 301,\ 56]$.

We note that the estimates, $\hat{\bar{\mathbf{P}}}_{cur}$ are less ambiguous, but also less credible than $\hat{\bar{\mathbf{P}}}_{cma}$ because they have only one sample at each $\mathbf{x}_k$ in 2(b). Therefore, the best estimate can be obtained by choosing the better one between two estimates for each appliance. For example, if the second appliance has non-stationary and others have stationary active power consumption, then the best estimates would be selecting the estimate of $\bar{P}_2$ from $\hat{\bar{\mathbf{P}}}_{cma}$ and others from $\hat{\bar{\mathbf{P}}}_{cur}$. So we have the optimal estimates of $\hat{\bar{\mathbf{P}}}_{opt} = [111.88,\ 57.7,\ 301]$.

As shown in the above example, our algorithm first computes two estimates of active average power consumptions; $\hat{\bar{\mathbf{P}}}_{cma}$ from the entire data set and $\hat{\bar{\mathbf{P}}}_{cur}$ from the partial data set. Then it selects the best between two estimates for each appliance. For the criteria for the selection the algorithm predict Mean Square Errors for each appliance for each estimate, $\hat{\bar{\mathbf{P}}}_{cma}$ and $\hat{\bar{\mathbf{P}}}_{cur}$.

Estimation Accuracy Prediction We can predict the Mean

Square Error by considering the variance of the estimated average power consumptions, $\hat{\bar{\mathbf{P}}}$. Then we can have the following equalities where the second equality comes from $E(\hat{\bar{\mathbf{P}}}) = \bar{\mathbf{P}}$ for unbiased errors, i.e $E(\mathbf{e}) = 0$

$$Var(\hat{\bar{\mathbf{P}}}) = E\left[\left(\hat{\bar{\mathbf{P}}} - E(\hat{\bar{\mathbf{P}}})\right)^2\right] = E[(\hat{\bar{\mathbf{P}}} - \bar{\mathbf{P}})^2]$$

The variance of a coefficient estimate is proportional to $(\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}$ where $\mathbf{X}$ is a data matrix from binary sensors and $\mathbf{W}$ is a weight matrix to be designed for better estimation [16]. Therefore, the MSE of average active power consumption of ith appliance, $MSE(\bar{P}_i)$ can be obtained by computing a ith diagonal element of $Var(\hat{\bar{\mathbf{P}}})$ as shown in (12).

$$MSE(\bar{P}_i) = \sigma_T^2[(\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}]_{ii} \tag{12}$$

where σ_T^2 is a variance of the total power consumption, $y(t)$, i.e $\sigma_T^2 = Var(e(t))$ where $e(t)$ is defined in (6).

An estimate of σ_T^2 is given by

$$\hat{\sigma}_T^2 = \frac{\sum_{t=1}^{n}(y(t) - \hat{\bar{\mathbf{P}}}_{\mathbf{k}}\mathbf{x}(t))^2}{n-(p+1)} \tag{13}$$

which is the residual sum of squares of $y(t)$ divided by its degree of freedom, $n-(p+1)$ where n is the number of samples and p is the number of appliances.

5.4 Algorithm Implementation

Algorithm 1 outlines the energy breakdown estimation algorithm for a single meter. At every t second($T_s = 1$), the algorithm collects a sample of the total power consumption, $y(t)$ from a power meter and updates the Weighted Mean Table, **TB** given the current binary state, $\mathbf{x}(t)$ (line 3). The kth row of the Table maintains a queue, $\mathbf{Q}_k$ which stores the sample, $y(t)$ given $\mathbf{x}(t) = \mathbf{x}_k$ and a pointer, ptr_k which indicates the first sample of the current estimation period.

If the current time, t is the end of the current estimation period, then the algorithm computes the energy breakdown (line 4). The algorithm estimates the variance of the active power consumption α from the weighed mean table **TB**. The average active power consumption is computed for a cumulative data set, $\bar{\mathbf{P}}_{\mathbf{cma}}$ and a current data set, $\bar{\mathbf{P}}_{\mathbf{cur}}$, and the optimal estimates, $\bar{\mathbf{P}}_{\mathbf{opt}}$ are selected by the metric in (12) (line 8-12). The energy consumption of ith appliance, $\hat{\mathbf{E}}(i)$ is simply computed by multiplying $\bar{\mathbf{P}}_{\mathbf{opt}}(i)$ by the ON state duration during the estimation period (line 13). Finally, the pointer, ptr_k is reset for the next estimation period (line 15).

6 Power Meter Deployment

In this section we describe the additional meter deployment algorithm and its implementation. Overall the algorithm consists of two stages. First, the algorithm runs a single power meter on the root estimating average power consumptions and learning data statistic as described in the previous section. In the second stage, the algorithm finds the optimal number of additional power meters and their locations to improve estimation accuracy. For this, we design a cost function to quantify two conflicting goals, the number of power meters and estimation accuracy.

Algorithm 1 Estimate $\hat{\mathbf{E}}$ from $\mathbf{X}$ and $\mathbf{Y}$ for T_{est}

Require: $Rank(\mathbf{X}) \geq p$
Ensure: $\|\mathbf{Y}(t) - \hat{\mathbf{Y}}(t)\| \approx 0$
1: Create Table $\mathbf{TB} = [\mathbf{W}|\mathbf{X}|\bar{\mathbf{Y}}]$
2: **for** each power meter sample, $y(t)$ **do**
3: Update a queue, $\mathbf{Q}_k$ in **TB** given $\mathbf{x}(t) = \mathbf{x}_k$
4: **if** $t\%T_{est} = 0$ **then**
5: Estimate α^2 by solving (11)
6: Estimate $\bar{\mathbf{P}}_{\mathbf{cma}}$ and $\bar{\mathbf{P}}_{\mathbf{cur}}$ by solving (8)
7: **for** $1 \leq i \leq p$ **do**
8: **if** $\widehat{MSE}(\bar{\mathbf{P}}_{\mathbf{cma}}(i)) < \widehat{MSE}(\bar{\mathbf{P}}_{\mathbf{cur}}(i))$ **then**
9: $\hat{\bar{\mathbf{P}}}_{\mathbf{opt}}(i) = \bar{\mathbf{P}}_{\mathbf{cma}}(i)$
10: **else**
11: $\hat{\bar{\mathbf{P}}}_{\mathbf{opt}}(i) = \bar{\mathbf{P}}_{\mathbf{cur}}(i)$
12: **end if**
13: **return** $\hat{\mathbf{E}}(i) = \hat{\bar{\mathbf{P}}}_{\mathbf{opt}}(i)\left(\sum_{\tau=t-T_{est}}^{t} x_i(\tau)\right)$
14: **end for**
15: Reset ptr_k
16: **end if**
17: **end for**

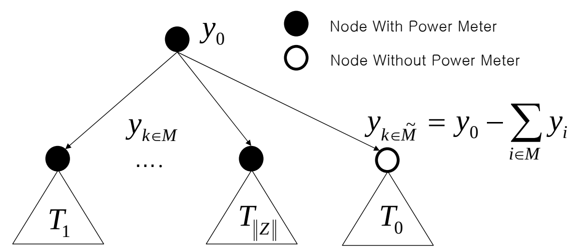


Figure 7. Tree abstract for power meter deployment

The power meter deployment is essentially a tree decomposition problem dealing with how to decompose the ambiguous data matrix, $\mathbf{X}$ of binary states into more apparent subsets of column vectors (or sub-trees). Figure 7 shows its tree representation where power meters are deployed on electrical outlets including one on the root. The electrical outlets can be partitioned into two groups with and without an power meter denoted by M and $\tilde{M}$ respectively. Assuming we have q electrical outlets we can define a state of power meter placement by a binary q tuple, $\mathbf{Z} = (z_1, z_2, ..., z_q)$, $z_i \in \{0,1\}$ where $z_i = 1$ if a power meter in at ith electrical outlet, and $z_i = 0$ otherwise. We have $Z = (0, 0..., 0)$ for a single power meter on the root.

Let $\mathcal{T}_i$ denote a subtree of appliances at ith power meter, $i \in \{0, 1, ...\|Z\|\}$. All appliances plugged into electrical outlets without a power meter are grouped into the subset, $\mathcal{T}_0$. The algorithm for decomposing the data matrix, $\mathbf{X}$ considers two conflicting objectives; the estimation accuracy, $\left\{\sum_{i=0}^{\|\mathbf{Z}\|} MSE(\bar{\mathbf{P}}|\mathcal{T}_i)\right\}$ and the number of power meters, $\|\mathbf{Z}\|$.

Decomposition Method The common way to optimize conflicting goals is to design a cost function and minimize it. As stated in section 4 involves two main challenges; how to predict the estimation accuracy at each subtree, $MSE(\bar{\mathbf{P}}|\mathcal{T}_i)$ and how to compute the optimal number of power meters and their locations avoiding a combinatorial explosion. The metric, $MSE(\hat{\bar{\mathbf{P}}})$ in (12) provides a good prediction for the estimation accuracy at each subtree. It requires only two statistics to predict estimation performance; the expected number of samples at each binary states and the variance of active power consumption of appliances. In order to avoid the search space explosion we compute the approximate solution instead of the exact solution using a stochastic optimization technique, Simulated Annealing [11].

To formulate the simulated annealing problem, we define a Markov Random Field on $\mathcal{Z}$. A Markov random field is a graphical model where the dependence among the random variables is determined by the edges of the graph through a generalization of the Markov property [7]. In our problem, a state vector, Z is represented as a vertices of the random field and state transition between vertices occurs by Markovian process. We say that two state nodes $s \in \mathcal{Z}$ and $t \in \mathcal{Z}$ are neighbors if and only if $d_1(s,t)$, 1-norm distance [1] between s and t is one. Let $nb(t)$ denote the set of neighbors of t, that is, $nb(t) = \{s \in \mathcal{Z}\, | d_1(s,t) = 1\}$. The goal of simulated annealing is to find an $i \in \mathcal{Z}$ minimizing $c(i)$, where c is a given cost function defined on the set of state nodes $\mathcal{Z}$ of the graph. The cost function is designed to quantify the trade-off between the predicted estimation accuracy, $MSE(\hat{\bar{\mathbf{P}}})$ and the number of power meters, $\|Z\|$ given a state node, $\mathbf{Z} \in \mathcal{Z}$. Both estimation quality and the number of meters are evaluated against the initial solution, denoted by $\mathbf{Z}_0$ that is placing meters on the all available electrical outlets.

Finally, we design the cost function as shown in (14) where $\lambda \in [0\ 1]$ is a weight coefficient and $\lambda = 0.5$ by default.

$$c(\mathbf{Z}) = \lambda \underbrace{\left\{ -\frac{MSE(\bar{\mathbf{P}}|\mathbf{Z_0})}{MSE(\bar{\mathbf{P}}|\mathbf{Z})} \right\}}_{estimaion\ quality} + (1-\lambda) \underbrace{\left\{ \frac{\|\mathbf{Z}\|}{\|\mathbf{Z_0}\|} \right\}}_{node\ efficiency} \tag{14}$$

In the cost function, the coefficient, λ quantifies the preference for estimation quality over the number of power meters. The initial solution, $\mathbf{Z_0}$ has the best estimation quality (or minimum $MSE(\bar{\mathbf{P}})$ but the worst node efficiency (or maximum $\|\mathbf{Z}\|$). The first term quantifies the estimation improvement over initial solution and its value ranges from -1 to 0. Similarly, the second term quantifies the node efficiency improvement over initial solution and its values ranges from 0 to 1. Therefore, the cost is set to 0 for initial solution.

Then simulated annealing evaluates the cost function over neighbors and make transition according to transition probability matrix. It runs a Markov chain, $\mathcal{Z}$ according to probability transition matrix, A_T defined by (15).

$$A_T(i,j) = \begin{cases} \frac{1}{d(i)} \left\{ \min(1, \frac{e^{-c(j)/T}}{e^{-c(i)/T}}) \right\} & \text{if } j \in nb(i) \\ 1 - \sum_{j \in nb(i)} A_T(i,j) & \text{if } j = i \\ 0 & \text{if } j \notin nb(i), j \neq i \end{cases} \tag{15}$$

where $d(i)$ is the number of neighbors of a state node i and $T > 0$ stands for temperature.

For each iteration step t, the temperature T exponentially decrease by the cooling schedule, $T(t) = T_0 \alpha^t$ where $0 < \alpha < 1$ and T_0 is a initial temperature. The iteration is stopped if the temperature becomes less than the equilibrium temperature, T_e. We note that the simulated annealing is a slow algorithm for this process, but it only needs to run once during the deployment. More efficient methods could also be applied, but we defer these as part of our future work.

7 Evaluation

To validate our algorithms we collected ground truth power consumption data from 12 appliances in a one-bedroom apartment for three days, from Thursday to Saturday using a commercial power meter, *Watts up.Net* [6]. The collected measurements, also including total consumption and state transitions were used to construct two case studies that we used to evaluate the accuracy of our algorithms. The optimization problems were solved using the *fmincon* function of MATLAB's optimization toolbox (8) and (11). We use this as a generic tool that can be easily substituted with other similar tools. The details of each case study are given below.

7.1 Case Study 1: A small electricity Network with single power meter

In this experiment, we evaluate the energy breakdown performance using collected power measurement data set of 12 appliances running in a one bedroom apartment. The actual power consumption of each appliance is collected by a power meter with $1Hz$ of sampling rate and its binary ON/OFF data of appliance is obtained by a simple thresholding. The hourly energy consumption profile of the 12 appliances is shown as a stacked bar graph in Figure 8. During the experiment, guests frequently visit and stay in the house and are involved in various activities such as cooking, watching TV, and playing games etc. A ceramic heater (in the bedroom) is pre-programmed to maintain the temperature to $75F$ thoughout day.

In order to understand the energy load characteristics, we summarize daily energy consumption profiles of appliances in Table 3. The Table shows a large variation of energy load among appliances, which is one of the challenges we identified for our problem. Furthermore, we note that the heater accounts for more than 60% of the total energy consumption due to the cold temperature outside during the experiment, $20 \pm 10F$. Conversely, the laptop consumed the least, less than 1% of the total load.

The collected data also exhibits the non-stationary property of active power consumptions. In Figure 9 shows a normalized histogram of power consumptions of appliances when binary sensor detects a state, *ON*. The Figure shows

[1]For a point $(x_1, x_2, \cdots, x_n)$ and a point $(y_1, y_2, \cdots, y_n)$, the p-norm distance is defined as: $d_p(x,y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}$

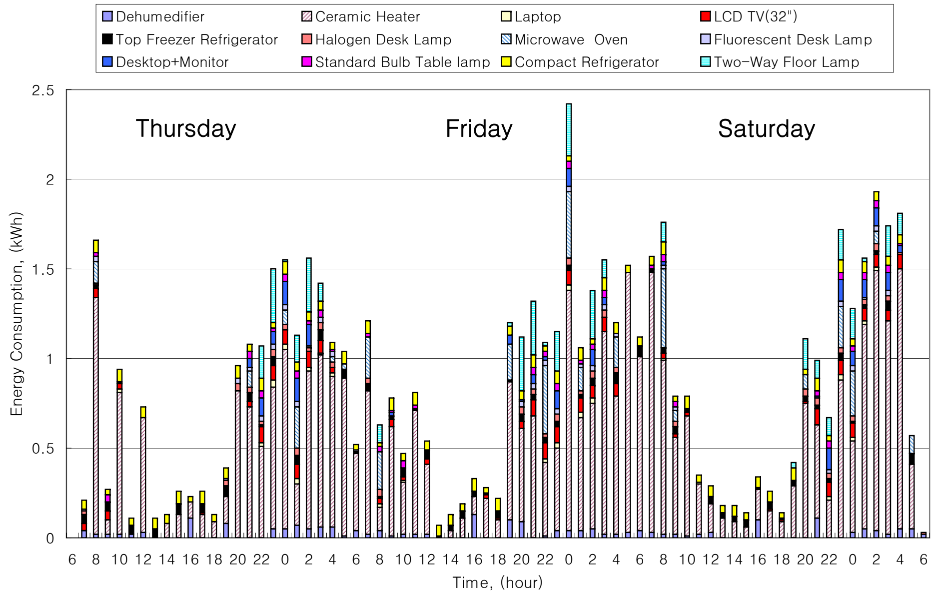


Figure 8. The hourly energy consumption ground truth in one-bedroom apartment from an experiment from Thursday to Saturday

Usage (kWh)	Dhm. -dfr.	Hea -ter	Lap- top	LCD TV	Top. Frg.	Hlg. Lmp.	Mcr. Oven	Fls. Lmp.	Dest -top.	Blb. Lmp.	Cmt. Frg.	Flr.2W Lmp.	Total kWh
Thur.	0.72	11.54	0.20	0.72	0.67	0.34	0.57	0.30	0.61	0.30	1.24	1.03	18.28
Fri.	0.74	13.31	0.19	0.84	0.71	0.36	1.68	0.26	0.48	0.33	1.24	1.62	21.80
Sat.	0.59	13.48	0.14	0.71	0.61	0.32	1.22	0.20	0.68	0.34	1.14	1.16	20.63
Average	0.68	12.78	0.17	0.75	0.67	0.34	1.16	0.25	0.59	0.33	1.21	1.27	20.24
Percent	3.4%	63.1%	0.8%	3.7%	3.3%	1.7%	5.7%	1.3%	2.9%	1.6%	5.9%	6.3%	100%

Table 3. Daily energy consumption breakdown of appliances in kWh and Percentage.

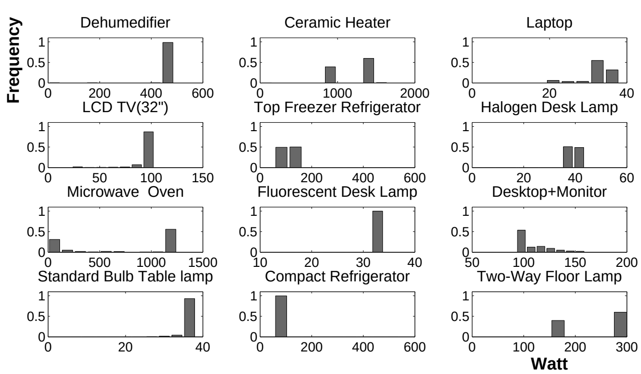


Figure 9. Histogram of power consumption of appliances during their On state (the frequency is normalized by the total number of events

the appliances, *Ceramic Heater*, *Microwave Oven*, and *Two-Way Floor Lamp* show two distinctive power modes while they are in state,*ON*. The computers, *Desktop+Monitor* and *Laptop* show more continuous power modes, and have an Idle mode at 100 Watt and 200 Watt respectively.

The total number of observed distinctive composite binary states is 336, which takes account for only 8%($=336/2^{12}$) of the maximum number of possible states. The occurrence of binary states shows a highly non-uniform distribution. The most frequently observed state is *when only compact refrigerator is on*, which takes account for 15.7% of the events among the total number of samples, 25920($= 24hrs \times 60min \times 60sec \times 3days$). The next most frequently observed states are *when all appliances are off* (7.1%), *when the compact refrigerator and the ceramic heater are on* (6.9%), *when the top freezer refrigerator and the ceramic heater are on*, and *when only ceramic heater is on*, (6.9%).

7.1.1 Hourly Energy Breakdown Estimation

We evaluate the estimation performance of our algorithm using the hourly energy consumption breakdowns. The ground truth is shown in Figure 8. We use the relative error as a percentage for the performance metric, which is defined by the ratio of the difference between the actual and estimated energy consumption to the actual, i.e

$$\textit{the relatve error} = 100 \times \frac{|\textit{the actual} - \textit{the estimated}|}{\textit{the actual enegry consumption}}$$

It is important to note that the metric is prone to large rounding error for vampire power, and so computations can be highly inaccurate. Therefore, the metric is evaluated only when an appliance is ON excluding OFF events from the performance evaluation.

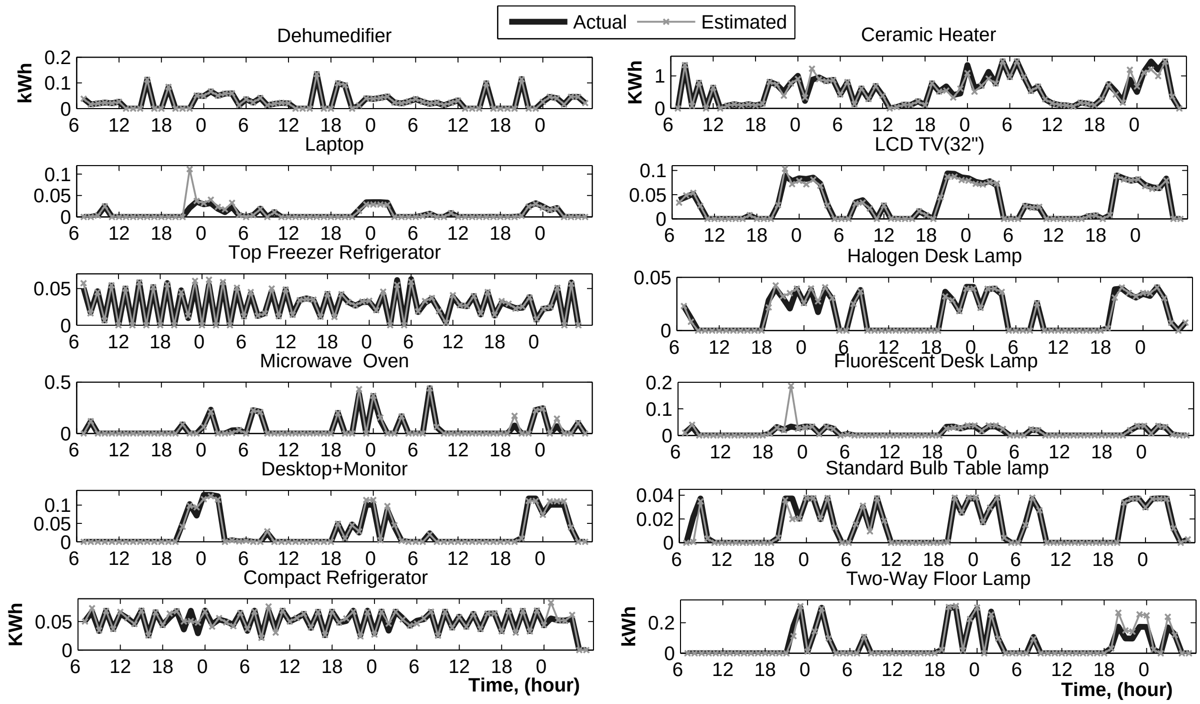


Figure 10. **Estimated hourly energy consumption profile (shown as a thin dotted line with a cross mark) of each appliance from Thursday to Saturday. The actual profiles are shown as a thick solid line**

The estimated hourly energy consumption profiles are plotted in Figure 10 as a thin dotted line with a cross mark, which compared to the actual profiles shown as a thick solid line. As shown in the figure, the estimated profiles closely follow the actual profiles. We note that there is a large estimation error for the *Laptop* and the *Fluorescent Desk Lamp* at 22 hrs (or 10 PM) on Thursday. This is because the algorithm estimates the active power consumptions with little cumulative data of those appliances upon estimation at 22 hrs, which results in a large variance of estimates. The Figure shows that no such large errors are observed after enough cumulative data is collected.

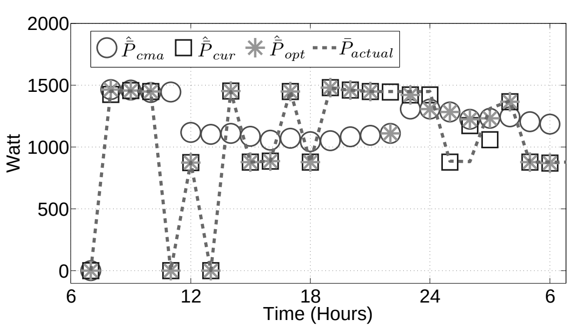


Figure 11. **The optimal data selection process for heater**

Figure 11, shows a more detailed snapshot of estimation process for the *Ceramic Heater* during the first day. The algorithm estimates the average active power consumptions

		actual optimal selection	
predicted optimal selection		current	cumulative
	current	520	42
	cumulative	130	172
total observed		650	214
error rate		20.00%	19.62%

Table 4. **The number of optimal data sets and the data selection error rate during the hourly energy consumption breakdown estimation**

using the entire cumulative data set, $\hat{\bar{\mathbf{P}}}_{\mathbf{cma}}$ (shown by a rectangular mark) and the data set during the current estimation period, $\hat{\bar{\mathbf{P}}}_{\mathbf{cur}}$ (shown by a circular mark). Then it selects a better (optimal) estimate, $\hat{\bar{\mathbf{P}}}_{\mathbf{opt}}$ (shown by a star mark). The actual average active power consumption is plotted as a dotted solid line in the figure. The Figure shows that $\hat{\bar{\mathbf{P}}}_{\mathbf{cur}}$ is closer to the actual average power consumption than $\hat{\bar{\mathbf{P}}}_{\mathbf{cma}}$, which is expected for the non-stationary active power consumptions. This demonstrates that the algorithm makes the right choice most of the time.

Table 4 shows the prediction performance for the optimal data selection for the hourly energy breakdown estimation. The total number of prediction trials is 864 ($= 3days \times 24hrs \times 12appliances$). The cumulative data and the current data are optimal for $75\%(650/864)$ and $25\%(214/864)$ of the total estimates, respectively. The prediction error is roughly 20% for both cases.

Table 5 summarizes the average relative error of hourly

Error (%)	Dhm. -dfr.	Hea -ter	Lap-top	LCD TV	Top. Frg.	Hlg. Lmp.	Mcr. Oven	Fls. Lmp.	Dest -top.	Blb. Lmp.	Cmt. Frg.	Flr.2W Lmp.	Avg.
Thur.	0.98	6.15	60.99	7.74	6.41	17.35	18.14	46.18	8.83	13.63	8.28	8.25	16.91
Fri.	1.05	5.77	10.86	4.48	3.59	4.04	5.31	14.53	9.96	7.23	5.01	2.94	6.23
Sat.	0.78	6.30	9.92	3.58	4.81	4.53	33.45	8.03	8.32	1.52	4.07	30.71	9.67
Avg.	0.94	6.07	27.25	5.27	4.94	8.64	18.97	22.91	9.04	7.46	5.79	13.97	10.94

Table 5. Average relative error of hourly energy consumption estimates for each day

energy consumption estimates per appliance for each day. It shows that the algorithm starts with a relatively large error in the first day, but the estimation performance is significantly improved in the second and third days. The overall average relative error of algorithm is 10.94%.

7.1.2 *Performance by Data Selection, Weight Matrix, and Estimation Period*

Next, we examine how the performance changes over different estimation periods (T_{est}), weight matrix ($\mathbf{W}$), data selections ($\hat{\hat{\mathbf{P}}}_{\mathbf{cma}}$ or $\hat{\hat{\mathbf{P}}}_{\mathbf{cur}}$). To evaluate the impact of weight matrix on performance, we use the following 4 different types of weight matrix.

- No Weight: $w_k = n_k$
- Unit Sum Matrix: $w_k = n_k / \sum_i \mathbf{x}_k(i)$
- Estimated Variance Sum Matrix: $w_k = n_k / \sum_i \hat{\alpha}_i^2 \mathbf{x}_k(i)$
- Exact Variance Sum Matrix: $w_k = n_k / \sum_i \alpha_i^2 \mathbf{x}_k(i)$

For comparison purposes, in addition to the proposed weight matrix, the *Estimated Variance Sum Matrix* we run our algorithm with the known exact variance, α^2, (*Exact Sum Variance Matrix*) which would give the best performance. The effect of using a poor weight matrix design is shown in *Unit Sum Matrix*. In the *No Weight*, the algorithm ignores the diversity of binary sensor data by giving an equal weight, 1 to all binary samples except for normalizing by the number of observed samples, n_k.

We also compare the estimation performance for the following 4 different data selection schemes: cumulative data set, current data set, predicted optimal data set, actual (or predicted by an oracle) optimal data set.

We compare the estimation performance of the proposed data selection scheme, *Predicted optimal data set* to the performance when using the entire cumulative data set only and when using the current estimation period data set only. We also evaluate the performance when the algorithm has a perfect predictor, i.e an oracle that would tell us which data set to choose. The best achievable performance (or the lower bound of estimation error) is obtained when we always select the optimal data set (Oracle Data Selection) with the exact variance of active power consumption of each appliance (Exact Variance Sum Weight). We compare our algorithm to the lower bound to examine how much we can further improve our algorithm.

We evaluate the relative error of average active power consumptions for each appliance and compute its average value given an estimation period, T_{est}. We observe how the average relative error changes when we increase the estimation periods from 5 min to 3 hours by $5min$. In Figure 12(a), we compare the estimation performance of our algorithm ($Est.Var.Sum.Wgt + Opt.Data.Sel$)to its lower bound ($Exact.Var.Sum.Wgt + Oracle.Data.Sel$).

relative error (%)	no weight	unit.var weight	est.var weight	exact.var weight
cma.data	66.19	62.43	17.51	8.16
cur.data	52.77	52.38	47.00	42.39
opt.data	42.91	40.80	**9.48**	5.91
oracle.data	23.07	21.34	5.50	3.32

Table 6. Average relative error of active power consumption for all estimation periods

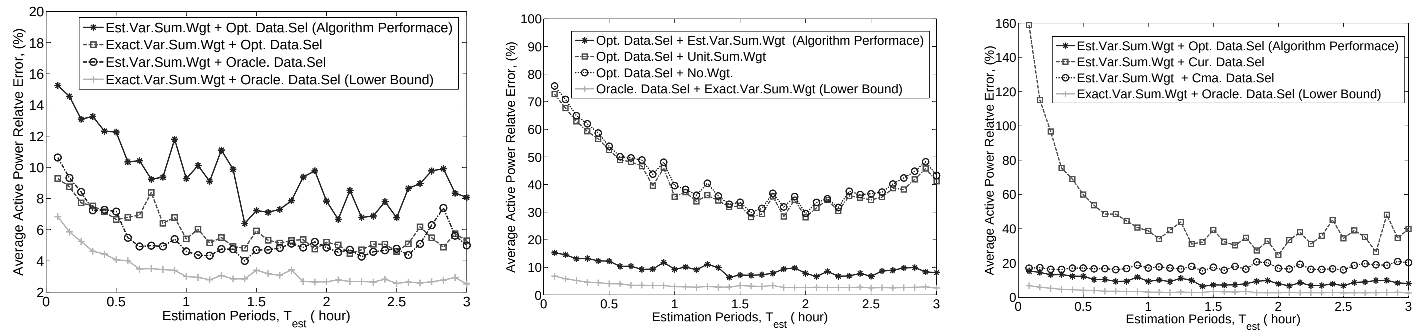
Our algorithm shows $10 \sim 15\%$ of average relative error in $5min \sim 1hrs$ of estimation periods and $6 \sim 10\%$ in $1hrs \sim 3hrs$. The figure shows that we can reduce the average relative error by $30 \sim 40\%$ with the known exact variance, ($Exact.Var.Sum.Wgt$) or perfect data selection ($Oracle.Data.Sel$). The relative error can be further reduced by more than 50% resulting $3 \sim 7\%$ of average relative error if we estimate with the known exact variance and the perfect data selection, which is the performance bound of our algorithm.

We further explore the impact of weight matrix and data selection scheme on estimation performance in Figure 12(b) and (c). Figure 12(b) shows that using *Unit Sum Matrix* or *No Weight* instantly degrades the estimation performance by $3 \sim 5$ times. In Figure 12(c) we can observe that the relative error tends to rapidly decrease when the *Current data set* used and slightly increase when *Cumulative data set* used for longer estimation periods. Note that the *Cumulative data set* shows similar performance with the predicted optimal data set by the metric ,(12) during the shorter estimation periods. The performance gap, however becomes widened for longer estimation periods as much as twice.

Finally, we summarize the result of relative error performance in Table 6 and in Figure 13. Table 6 shows the average relative error for all estimation periods and appliance with all possible combination of weight matrix and data selection schemes. Our algorithm shows 9.46% of relative error on average, which can be potentially reduced to as low as 3.32%. Figure 13 shows the maximum, minimum, and average value of relative error. We note that the worst case performance (the maximum relative error) is rapidly reduced by using our algorithm, specially compared to average relative error.

7.2 Case Study 2: A large scale electricity network with electrical meter deployment

In this section we present simulation results of our algorithm for a large scale electricity network of a large building. In the simulation, ON and OFF events are generated



(a) (b) (c)

Figure 12. the average relative error of the active power consumption over estimation periods (a) with the *Oracle Data Selection* or/and the *Exact Variance Sum Matrix*, (b) with different weight matrix given the *Predicted Optimal Data Set*, (c) with different data selection schemes given the *Estimated Variance Sum Matrix*

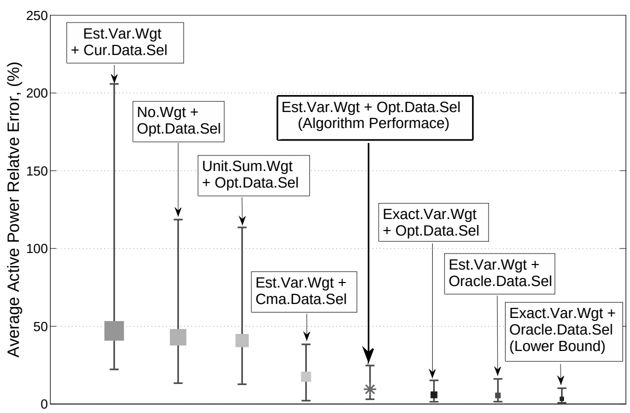
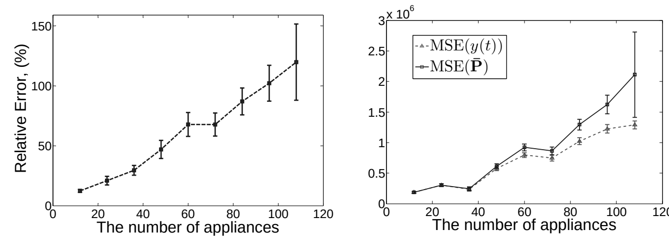


Figure 13. The maximum, minimum, and average value of relative error of active power consumption for all estimation periods with various combination of weighted matrix and data selection schemes

by Semi-Markov Chain where each appliance independently stays either of ON or OFF state for random amount of time with exponential distribution. The duty period and duty cycle of each appliance are generated independently by the uniform distribution of $[1hr-6hr]$ and $[0-1]$ respectively. To make the simulation more realistic, we use the power consumption profile data collected from the previous experiment.

We increase the number of units in a building from 1 to 9 assuming that each unit has 12 appliances, which are randomly selected from 12 appliances used in previous experiment. Therefore, the total number of appliances on the electricity network in the building increases from 12 to 108 by 12. Then we run our algorithm with a single power meter on the root for 3 days with 1 hour as the estimation period. We run the simulation for 100 times at each number of appliances, by which we can test our algorithm for various sets of ON/OFF binary event profiles. Then we compute the average estimation performance over the number of appliances (or units in a building).



(a) (b)

Figure 14. Performance evaluation with a single power meter for a large electricity network: a) the relative error of the average active power consumption of appliance, b) the performance in $MSE(\mathbf{P}(t))$ and $MSE(y(t))$

Figure 14(a) shows the relative error of the average active power consumption. The relative error per appliance and its variance grows almost linearly with the number of appliances. In the figure the relative error reaches 50% for 48 appliances and exceed 100% for 96 appliances. We also evaluate the MSE performance of the average active power consumption, $MSE(\bar{\mathbf{P}})$ and the total power consumption, $MSE(y(t))$ in Figure 14(b). We note that the performance of $MSE(\bar{\mathbf{P}})$ is more affected than the $MSE(y(t))$ by the number of appliances. This is because the ON/OFF binary sequences are more likely correlated as the number of appliances increases.

Then we evaluate the performance of the meter deployment algorithm for a building with 12 units where each unit has a set of randomly assigned 12 appliances, i.e 144 appliances on the electricity network. Each unit is assumed to have one electrical outlet for meter deployment. Therefore, the total number of meters is from 1 to 13 including the meter on the root. We compare our deployment algorithm to a random deployment, which simply choose a random number of additional power meters from $\{0, 1, \cdots, 12\}$ and randomly place them at electrical outlets. By default, at least one meter is on root.

We run the random deployment for 1000 times and plots the number of power meters and corresponding estimation performance, $MSE[\bar{\mathbf{P}}]$ in Figure 15. At each number of

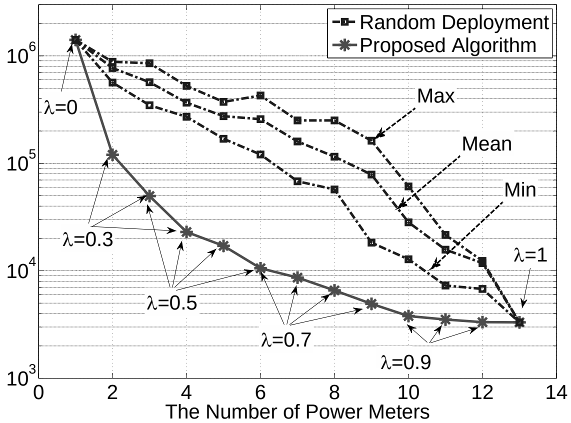


Figure 15. The performance of meter deployment algorithm vs random deployment

power meter the minimum, the average, and the maximum value of $MSE[\bar{\mathbf{P}}]$ are shown in the figure. It shows the clear trade-off between the number of power meters and the estimation performance with a large variance. The performance of our algorithm is compared to the random deployment in the same plot. We run the algorithm with different weight coefficients, $\lambda = \{0, 0.3, 0.5, 0.7, 0.9, 1\}$ of cost function in (14). For each coefficient value, we run the algorithm for 5 times. It shows a great improvement in trade-off compared to the random deployment. For default value $\lambda = 0.5$ the algorithm deploys $3 \sim 6$ meters with topologies which provides 10 times better performance than random deployment. In other hands, it reduces the number of power meters by $2 \sim 3$ times compared to the random deployment given the same estimation performance. For two extreme cases, $\lambda = 0$ and $\lambda = 1$ the algorithm deploy no additional meter and meters on all outlets.

8 Conclusions and Future Work

In this paper we provided a method for estimating the power consumption breakdown per appliance inside a home using simple ON/OFF appliance state information. Our evaluation using actual measurements demonstrated that the close consideration of weights and dataset selection can play a key role in enhancing breakdown estimates. Our evaluation also showed that the proposed method scales well with the number of appliances. According to our simulation results, our incremental deployment algorithm for additional meters in large buildings can help reduce the number of meters needed by two to three times when compared to a more random deployment approach. Our evaluation also verified that our estimation performance metric works well and was successfully applied to demonstrate for a large electricity network with 144 appliances and 12 electrical outlets. The simulation result suggests that our algorithm can improve estimation performance by 10 times or reduce the number of electricity meters by $2 \sim 3$ times. In the near future, our plan is to deploy our system in a residential college on our campus, to provide energy breakdown estimates as feedback to the students. Another deployment in one of the engineering buildings will use our system to estimate the consumption of air handling controls for different floors with the ultimate goal of enabling the building to provide demand/response services by being able to make decisions based on our breakdown estimates. We will also extend our algorithm to leverage user inputs to do the same estimation for a partial set of appliances eliminating the need to have state information from all appliances.

9 References

[1] Arch rock. `http://http://www.archrock.com/`.

[2] The energy detective (ted). `http://www.theenergydetective.com/`.

[3] Millennial net. `http://www.millennial.net/`.

[4] Sensicast. `http://www.sensicast.com/`.

[5] The tendril residential energy ecosystem (tree). `http://www.tendrilinc.com/`.

[6] Watts up.net. `http://www.wattsupmeters.com/`.

[7] R. Chellappa and A.K. Jain. *Markov random Fields, theory and applications*. Academic Press, Boston, 1993.

[8] G.W Hart. Nonintrusive appliance load monitoring. In *roceedings of the IEEE*, December 1992.

[9] X. Jiang, M. Van Ly, J. Taneja, P. Dutta, and D. Culler. Experiences with a high-fidelity wireless building energy auditing network. In *SenSys '09: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, 2009.

[10] Y. Kim, T. Schmid, Z. M. Charbiwala, and M. B. Srivastava. Viridiscope: design and implementation of a fine grained power monitoring system for homes. In *Ubicomp '09*, 2009.

[11] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science, Number 4598, 13 May 1983*, 220, 4598:671–680, 1983.

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

[13] J. Lifton, M. Feldmeier, Y. Ono, C. Lewis, and J. A. Paradiso. A platform for ubiquitous sensor deployment in occupational and domestic environments. In *IPSN '07*, 2007.

[14] 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), 1996.

[15] S. N. Patel, T. Robertson, J. A. Kientz, M. S. Reynolds, and G. D. Abowd. At the flick of a switch: Detecting and classifying unique electrical events on the residential power line. In *UbiComp '07*, 2007.

[16] S. Weisberg. *Applied Linear Regression*. Wiley, thrid edition, 2005.