

Tracking States of Massive Electrical Appliances by Lightweight Metering and Sequence Decoding *

Yongcai Wang[†], Xiaohong Hao[†], Lei Song[†],
Chenye Wu[†], Yuexuan Wang[†], Changjian Hu[‡], and Lu Yu[‡]

[†] ITCS, IIS, Tsinghua University, Beijing, P.R.China

[‡] NEC Labs., Beijing, P.R. China

ABSTRACT

To smartly control the massive electrical appliances in buildings to save energy, the real-time on/off states of the electrical appliances are critically required as the fundamental information. However, it is generally a very difficult and costly problem, because N appliances have 2^N states and the appliances are massively in modern buildings. This paper propose a novel compressive sensing model for monitoring the massive appliances' states, in which the sparseness of on/off switching events within a short observation interval is exploited. Based on such a temporal sparseness feature, a lightweight state tracking framework is proposed to track the on/off states of N appliances by deploying only m smart meters on the power load tree, where $m \ll N$. Particularly, it firstly presents an online state decoding algorithm based on a Hidden Markov Model of sparse state transitions. It reduces the traditional $O(t2^{2N})$ complexity of Viterbi decoding to polynomial complexity of $O(tn^{U+1})$ where $n < N$ and U is an upper bound of the simultaneous switching events. To minimize the meter deployment cost, i.e., m , an entropy-based necessary condition for deploying the minimal number of meters while guaranteeing the state tracking accuracy is presented. Based on it, a greedy algorithm to optimize the meter deployment to meet any given state decoding accuracy requirement is proposed. These proposed results are verified extensively based on the simulated data and the real PowerNet data. Simulation results confirm the the good performances of the proposed methods, and also demonstrate some interesting structures of the problem.

Keywords

Energy auditing, Smart meter, Deployment optimization, State Tracking, Compressive sensing, Energy saving, Smart building

*This work was supported by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301, the National Natural Science Foundation of China Grant 61033001, 61061130540, and 61073174. Corresponding author e-mail: wangyc@tsinghua.edu.cn; Tel.: 86-10-62795843-1682.

[†] The first five authors are with Institute for Theoretical Computer Science, Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, 100084, P.R. China.

[‡] The last two authors are with NEC Labs. Beijing, 100084, P.R.China

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SensorKDD '12, August 12, 2012, Beijing, China.

Copyright 2012 ACM 978-1-4503-1554-8/12/08 ...\$10.00.

1. INTRODUCTION

Recent survey shows that in our offices, up to 70% of computers and related equipments are left on all the time [1]. To reduce the energy waste caused by such idle running, the real-time on/off states of the electrical devices are required as the necessary state information for smart control technologies. But tracking the real-time on/off states of the appliances is a challenging problem, because the appliances are massive and widely distributed in buildings. Traditional fidelity energy monitoring systems generally require large-scale smart meter networks, and thus suffer from the high deployment, maintenance and data collection costs. In real applications, it is always desirable to design a low-cost and efficient method to track the states of the massive appliances.

The on/off duration of an appliance is a critical variable to estimate the appliance's power consumption, because the energy consumption pattern (how much energy an appliance consumes when it is on) can be learned off-line from the nominal power data sheet or by an off-line learning process [2][3]. Therefore, in this paper, we consider deploying m smart meters to track the on/off states of N appliances in real-time, where $m \ll N$. Particularly, the power distribution network in a building has a typical tree structure, where the leaf nodes are the appliances. Smart meters may be deployed at the root (power entrance), at some intermediate nodes (power switches or outlets), or at the leaf nodes of the tree. Each meter monitors the aggregated current/power consumption of the electrical appliances in the subtree rooted at itself. The goal is to disambiguate the on/off states of the N appliances from the mixed power measurements of the m meters.

1.1 Related Work

The energy auditing and monitoring problem has caught tremendous attentions from both academia and industry for the last decade. There are three main bodies of the related literature.

1. Bottom up monitoring approach. The first category focuses on designing smart meter network for detailed energy monitoring. An early work is the MIT Plug system [4], where the design and development of the smart metering system were reported with a trial deployment of 35 smart meters on a floor of a building. Jiang *et al.* [5] reported the design and development of Berkeley AC meter network exploiting the idea of web of things. The same authors reported utilizing contextual metadata for the high-fidelity monitoring and spatial, functional, and individual decomposition of electric usage in buildings in [6]. A recent work by Dawson-Haggerty *et al.* [7] shared their insights obtained from a year-long, 455 meter deployment of wireless plug-load electric meters in a large commercial building. In [8], Kazandjieva *et al.* introduced PowerNet,

which was a hybrid sensor network for monitoring the power and utilization of computing systems in a large academic building. Jung *et al.* [9] proposed energy breakdown research with consideration of minimizing meter deployment cost, but they assume the appliance's on/off states are sensed by additional RFID sensors. There are also solutions from the industry, such as Tendril [10], Greenbox [11], EnergyHub [12], and Sensicast [13]. In contrast to the existing work, this paper in the first time studies the impact of the meter deployment to the decoding accuracy, so that the meter deployment cost can be saved remarkably while guaranteeing highly accurate state monitoring by sequential decoding.

2. Top down disaggregation approach. The second category focused on the non-intrusive load monitor (NILM) based on/off state disaggregation. In particular, the NILM-based method can efficiently reduce the deployment cost for smart metering, since it only deploys one high-frequency smart meter at the root of the power load tree to disambiguate the on/off states of the appliances by transient or static signal processing and pattern recognition. The first NILM approach was proposed by Hart in [14], which used real and reactive power measurements to detect the specific load signatures of individual appliances. Norford *et al.* [2] and Leeb *et al.* [3] proposed transient event detection methods to analyze the specific patterns in the spectral domains. Patel *et al.* [15] tried to recognize the electrical noise on the residential power lines to detect the on/off switching events. Farinaccio *et al.* [16] proposed a method to disaggregate the total electricity consumption into the major end-uses by pattern recognition. However, the NILM-based approach generally need the high-cost meter for high frequency sampling, and thus is limited in the application scale. To tackle this challenge, in this paper, we propose a lightweight approach which can monitor a load tree connecting massive appliances and has less requirements to the meter device.

3. On/off detection by additional sensors. Instead of utilizing smart meters, the final body concerned the on/off states monitoring by other types of sensors. For example, Kim *et al.* developed ViridiScope system [17], which detected the ambient signals emitted from appliances to infer the power consumption of appliances. Gupta *et al.* [18] proposed ElectriSense, which sensed EMI (electromagnetic interference) by a single point sensing for electrical event detection and classification in the home. Rowe *et al.* [19] used contactless sensing to monitor the variations in electromagnetic fields. Taysi *et al.* [20] proposed Tinyears to utilize audio sensor nodes.

A notable character in this problem is that the on/off states of an appliance are highly correlated in the time domain, which is rarely exploited in the previous studies. When an appliance is turned on, it generally works a long time before it is turned off and an appliance must be turned off (on) before it can be turned on (off). So that, for N appliances, their on/off switching events are quite sparse during a short observation period (for example a second), and the on/off events of an appliance must happen in turn. We call these two features: *switching sparseness* and *sequence feasibility constraint*. Thus, the problem can be converted to a problem of sparse switching event detection, by $m \ll N$ meters sampling in continuous short intervals. A typical sampling interval could be of length one second.

1.2 Our Contributions

However, even utilizing the temporal sparseness characters, to decode 2^N combinatorial states from the readings of m meters is still

difficult. Ambiguities and inefficiency are the main challenges because the problem is essentially a NP-complete sub-set sum problem. For a measurement instance of m meters at time t , there could be multiple states of the N appliances that have the same energy consumptions in terms of the smart meter measurement. Determining the right state from the ambiguities is difficult, and the state decoding complexity is $O(t2^{2N})$ in the traditional Viterbi decoding. Note that the disambiguation difficulty is mainly determined by the meter deployment scenario. A meter can disambiguate the states of its monitoring appliances successfully only if these appliances have no ambiguous combinatorial states. But deploying more meters will increase the cost of the metering system. Then, how should we deploy the minimal number of smart meters while guaranteeing no ambiguity? The focus of this paper is to answer this question, and the major contributions of this paper could be summarized as follows:

- To achieve an efficient and correct decoding, we propose a hidden Markov model (HMM) based decoding method to embed the switching sparseness and the sequence feasibility constraint into the decoding process. It recovers the on/off state path of N appliances from the sequential readings of m meters from time 1 to t . We further design a fast sequence decoding (FSD) algorithm, which exploits the idea of the off-line load-tree splitting, the state vector pre-ordering in each split tree, and the online state sequence likelihood ranking by forward and backward searching. The FSD algorithm runs in parallel in each split tree, with polynomial time complexity $O(n^{U_t+1})$, where $n < N$ is the number of appliances in a split tree, and U_t is the upper bound of simultaneous switching events in a split tree in a sampling slot.
- We investigate the relationship between the sensor deployment scenario and the state decoding accuracy by an entropy-based analysis. We propose the notion of "clear ratio" as a bridge to connect the tracking accuracy and the deployment cost. A meter deployment optimization algorithm (MDOP) is proposed based on the idea of entropy maximization to give a near-optimal, adjustable deployment strategy based on the requirement to the tracking accuracy.
- Extensive evaluations based on the simulated data and the real PowerNet data were carried out to show the efficiency, correctness and deployment cost saving performances of the proposed decoding and deployment optimization methods.

The rest of this paper is organized as follows. We introduce the system model in Section II. The HMM-based efficient sequential decoding method is introduced in Section III. Deployment optimization algorithm is presented in Section IV. Evaluation results will be presented in Section V, and the paper is concluded with discussion of future work in Section VI.

2. SYSTEM MODEL

The energy distribution network in a building has a typical tree-like structure [6]. The root of the load tree is the main power entrance of the building. Each node in the middle tier is a power break or an outlet and the leaf nodes of the tree are the electrical appliances. In the load tree, the power consumption at a node is equal to the sum of the power consumptions of the appliances in the subtree rooted at the node. Smart meters can be deployed at any node in the tree.

1. Observation Model

When m meters are deployed on the load tree to track the on/off

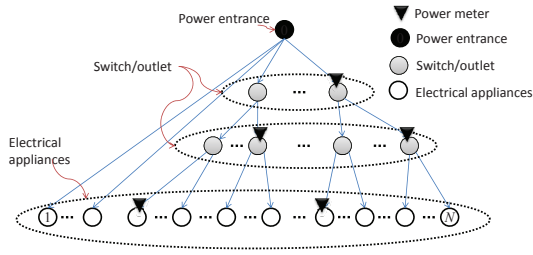


Figure 1: Sparse meter deployment scenario on the load tree.

states of N appliances. We can imagine the meter deployment scenario as shown in Fig.1. Each meter measures the real-time aggregated power of the appliances in the subtree rooted at itself. We assume all the meters are synchronized. At time t the observation model of a meter i can be formulated by:

$$\mathbf{Z}_{t,i} = \sum_{j \in S(i)} \mathbf{X}_{t,j} \mathbf{P}_{t,j} \quad (1)$$

where $S(i)$ is the subtree rooted at the meter i , $\mathbf{X}_{t,j} \in [0, 1]$ is the state of appliance j at time t , and $\mathbf{P}_{t,j}$ is the real-time power consumption of appliance j at time t . The goal of state decoding at meter i is to find the state vector of the appliances $\{\mathbf{X}_{t,j}, j \in S(i)\}$ to minimize the following expected square error:

$$\underset{\mathbf{X}}{\text{minimize:}} \mathbb{E} \left\{ \left(\mathbf{Z}_{t,i} - \sum_{j \in S(i)} \mathbf{X}_{t,j} \mathbf{P}_{t,j} \right)^2 \right\}. \quad (2)$$

By assuming $\mathbf{P}_{t,j}$ is a random variable with mean \mathcal{P}_j and variance $\delta_j = \alpha_i \mathcal{P}_j$, we prove in Appendix that the solution of problem (2) is approximately the same as the following problem:

$$\underset{\mathbf{X}}{\text{minimize:}} \left(\mathbf{Z}_{t,i} - \alpha/2 - \sum_{j \in S(i)} \mathbf{X}_{t,j} \mathcal{P}_j \right)^2, \quad (3)$$

where α is defined as the expectation of variance/mean ratio of appliances power consumption patterns, i.e., $\alpha = \mathbb{E} \{\delta_j / \mathcal{P}_j\}$.

2. Pattern Matrix

Based on (3), we denote $\mathbf{Y}_{t,i} = \mathbf{Z}_{t,i} - \alpha/2$ and consider the state decoding problem of the N appliances on the load tree, which can be formulated by (4).

$$\underset{\mathbf{X}}{\text{minimize:}} \|\mathbf{X}_t \mathbf{P} - \mathbf{Y}_t\|_2 \quad (4)$$

In (4), $\mathbf{X}_t \in (0, 1)^N$ is the state vector, indicating the on/off state of the N appliances at time t . $\mathbf{Y}_t \in \mathbb{R}^m$ is called the observation vector of the m meters. Different from the single meter case, a *pattern matrix* is introduced in (4) to model the meter deployment scenarios. When a meter deployment scenario is known, we can easily judge whether an appliance is in the subtree of a meter or not. We assume the load tree is static, so that the pattern matrix can be constructed from a given meter deployment scenario as follows:

1. The j th column of \mathbf{P} indicates the j th meter.
2. $\mathbf{P}_{i,j} = \mathcal{P}_j$, if appliance d_i is in the subtree of meter j ; otherwise $\mathbf{P}_{i,j} = 0$.

Problem (4) is to decode a N dimension variable \mathbf{X}_t from m dimension observation \mathbf{Y}_t . Since $m \ll N$, the number of equations is much less than the number of unknowns. Thus, there will be serious ambiguities in calculating \mathbf{X}_t , making the accurate state decoding very difficult.

3. On/off Switching Event Detection Model

An important fact to make the above model solvable is the temporal sparseness feature of the on/off switching events. A linear transformation is applied to \mathbf{X}_t , and \mathbf{Y}_t to obtain a new observation model:

$$(\mathbf{X}_t - \mathbf{X}_{t-1}) \mathbf{P} = (\mathbf{Y}_t - \mathbf{Y}_{t-1}), \quad (5)$$

where $\mathbf{X}_{t,i} - \mathbf{X}_{t-1,i}$ indicates the on/off switching event of the i th appliance from $t-1$ to t . $\mathbf{Y}_{t,j} - \mathbf{Y}_{t-1,j}$ is the measured power variation from $t-1$ to t at meter j . This transformation converts the state tracking problem to an on/off switching event detection problem. $\mathbf{X}_t - \mathbf{X}_{t-1}$ indicates the simultaneous on/off switch events happened in interval t , which is sparse when sampling interval is short. We denote U_t as the upper bound of the non-zero switching events in sampling interval t , i.e., $\|\mathbf{X}_t - \mathbf{X}_{t-1}\|_1 \leq U_t$, and $U_t \ll N$. So that, the decoding problem of \mathbf{X}_t by the meter measurements \mathbf{Y}_t can be solved by Least Squares Estimation (LSE) with a L1-Norm constraint, which is also known as the constraint type LASSO (Least Absolute Selection and Shrinkage Operator) problem[21].

$$\begin{aligned} &\text{minimize: } \|\mathbf{X}_t \mathbf{P} - \mathbf{Y}_t\|_2 \\ &\text{subject to: } 1. \|\mathbf{X}_t - \mathbf{X}_{t-1}\|_1 < U_t \\ &2. \forall i, \forall t, \mathbf{X}_{t,i} \in [0, 1]. \end{aligned} \quad (6)$$

Problem (6) can be solved by existing algorithms such as Tibshirani algorithm [21]. However, decoding \mathbf{X}_t only by the smart meter measurements at time t tends to be inaccurate because the information get at time t is very limited. It has not fully utilized the historical observations and the sequence feasibility constraint of the state sequence.

4. HMM-based State Sequence Decoding Model

Note that the state transition of electrical appliances has Markovian property, i.e., an appliance's state at interval t is only related to its state at interval $t-1$. Thus the state sequence decoding problem can be efficiently modeled by a Hidden Markov Model. The state space has size 2^N , which includes possible states of N appliances, denoted by S_1, \dots, S_{2^N} . The observation space contains all the possible distinct observations that maybe observed by the m meters, which is denoted by $V = \{v_1, v_2, \dots, v_M\}$. So that, the HMM model for state sequence decoding is formulated as $\lambda = (\mathbf{X}_0, \mathbf{A}_t, \mathbf{B})$, where \mathbf{X}_0 is the initial state distribution; \mathbf{A}_t is the state transition matrix, where $a_{i,j,t} = P(\mathbf{X}_t = S_j | \mathbf{X}_{t-1} = S_i)$, $i, j \in [1, \dots, 2^N]$ is the state transition probability from S_i at $t-1$ to S_j at t ; \mathbf{B} is the observation matrix, where $b_{i,j} = P(\mathbf{Y}_t = v_i | \mathbf{X}_t = S_j)$ is the likelihood of state S_j when the observation is v_i . Based on the HMM model, the problem of state sequence decoding can be formulated as following:

PROBLEM 1. Given the sequences of power measurements by m meters from time 1 to t : $\mathbf{Y} = \{\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_t\}$, and the HMM model λ , we want to find the state sequences of N appliances, $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_t\}$ that maximize the following conditional probability:

$$\begin{aligned} &\delta_t = \underset{\mathbf{X}_1, \dots, \mathbf{X}_t}{\text{maximize}} P(\mathbf{X}_1, \dots, \mathbf{X}_t, \mathbf{Y}_1, \dots, \mathbf{Y}_t | \lambda) \\ &\text{subject to: } 1. \forall \tau \in [1, t], \|\mathbf{X}_\tau \mathbf{P} - \mathbf{Y}_\tau\|_2 < \varepsilon, \\ &2. \forall \tau \in [2, t], \|\mathbf{X}_\tau - \mathbf{X}_{\tau-1}\|_1 < U_\tau, \\ &3. \forall i, \forall \tau, \mathbf{X}_{\tau,i} \in [0, 1], \end{aligned} \quad (7)$$

where ε is the tolerable measurement error of smart meters.

5. Building HMM of Sparse State Transitions

Training the HMM model is a critical step before using HMM to

decode the state sequence, which generally requires considerable training efforts, because the electrical appliances are massive. We propose to use lightweight off-line knowledge to build the HMM model and focus on the online decoding algorithm.

1. We propose to set up the transition matrix \mathbf{A}_t by the knowledge of U_t . Assume the on/off transition probabilities are equal and i.i.d. for N appliances, and denote p as the occupancy probability of one state change. Then p can be calculated by $\sum_{i=1}^{U_t} p^k (1-p)^{N-k} \binom{N}{k} = 1$. If d is the number of different states between S_i and S_j , i.e., $d = |S_i - S_j|_1$, then the transmission probability from S_j to S_i is modeled by $\mathbf{a}_{i,j,t} = p^d (1-p)^{N-d}$.
2. Modeling the observation matrix is generally difficult, because the distinct measurements measured by m meters are numerous. We propose to use a Least Square Estimation based online searching scheme to replace the explicit observation matrix model. When \mathbf{Y}_t is observed by m meters, a fast search algorithm will be executed to infer the hidden states that most likely generate the observations. It eliminates the efforts of training observation matrix and efficiently speeds up the online state decoding process.
3. The initial state of the HMM can be set by the off-line knowledge of the appliance states. Accurate knowledge on the initial states will improve the tracking accuracy. For example, the tracking algorithm can be started at mid-night, so that most appliances are in off state, which can be set as the initial states.

Therefore, the HMM model can be set up with very limited off-line knowledge, making it practical in state decoding.

3. FAST SEQUENCE DECODING

The HMM model in (7) has 2^N states and the sequence length is t . Traditional Viterbi decoding algorithms need $O(t2^{2N})$ complexity to decode the most likely hidden state sequence, which is hard to calculate in real-time when N is large. Further, in Viterbi algorithm, the reward of a state sequence is evaluated by the summation of vertex rewards and the link rewards associated to the sequence[22]. Since the vertex reward is affected by the measurement errors of the meters and the link rewards are determined by the state transition probabilities, these two rewards are not given from the same metric system. It is generally difficult to design weight for balancing them or determining which reward is dominant.

To solve these problems, this work exploits the tree structure of power network and the sparseness of on/off switching events to present a fast sequence decoding algorithm (FSD). The algorithm runs in parallel in the sub-trees of the load network, which has polynomial complexity $O(n^{U_t+1})$, where $n < N$ is the number of appliances in a sub-tree. It guarantees efficient real-time decoding even N is very large. Further, in FSD, we models both the link reward and the vertex reward by the occupancy probabilities of the corresponding event, and propose a product-type reward function instead of sum-type to rank the state sequence by overall occupancy probabilities. We show such reward functions are easily to set up and provide better decoding accuracy than traditional Viterbi.

1. Transform Load Tree to Mono-meter Tree Forest

We can decompose the state decoding problem by load tree splitting. For a node v in the load tree, if it is monitored by a smart meter, the power consumptions of the appliances in its subtree $ST(v)$ will be continuously monitored by the smart meter, so that all the

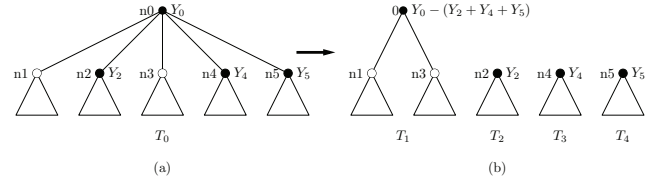


Figure 2: Load-tree splitting example

parents of v can know the power consumptions of its subtree. So that the subtree $ST(v)$ can be split from the full tree T . For a load tree $T = (V, E)$ with m meters deployed on it, it can be split into a forest of m mono-meter trees, where each mono-meter tree has only the root equipped by a meter while the others are not. A load tree T is transformed to a forest F by repeating following three steps on every tree in F :

1. For any tree T_0 in F rooted at n_0 , run Bread-first Search to find the first meter-equipped node n_1 s.t. $n_1 \neq n_0$;
2. Split T_1 , i.e., the subtree rooted at n_1 from T_0 and add T_1 into F ;
3. Subtract meter reading value of n_1 from reading of n_0 , i.e., $\mathbf{Y}_{t,0} \leftarrow \mathbf{Y}_{t,0} - \mathbf{Y}_{t,1}$;

This algorithm is initialized by assigning T to T_0 and is stopped when no tree in F can be further split, which ensures that every tree in F is a *mono-meter* tree.

Figure 2(a) shows an example load tree monitored by several meters. The leaf nodes stand for appliances while the others stand for outlets. The nodes equipped with smart meter are represented by black node and the nodes without smart meter are in white. Figure 2(b) shows the forest of mono-meter trees which is transformed from Figure 2(a).

2. Off-line State Sorting in Each Mono-meter Tree

Based on the tree splitting results, the online sequence decoding problem is decomposed to sequence decoding problem in each mono-meter tree. Considering a mono-meter tree with n appliances, we denote $\mathbf{y}_t = \{y_1, y_2, \dots, y_t\}$ as the observation sequence measured by the smart meter; denote $\mathbf{p} \in \mathbb{R}^{n \times 1}$ as the pattern matrix of the mono-meter tree and denote $\mathbf{x}_t = \{x_1, x_2, \dots, x_t\}$ as the state sequence of n appliances.

Note that the a mono-meter tree with n appliances still have 2^n possible states. Finding the feasible states that most likely generate the smart meter's observation at time t need $O(2^n)$ comparisons. For speeding up this step in the online phase, we off-line sort the 2^n states according to their energy consumptions to prepare an ordered state vector for efficient online binary search. Although the sorting operation has complexity $O(2^n \log(2^n)) = O(n2^n)$, its needs only to be executed once in off-line or infrequently in case of mono-meter tree structure changing.

3. Sequence Decoding Model in HMM Graph

After the mono-meter tree splitting and the off-line state sorting in each mono-meter tree, we now present the online state decoding algorithm in each mono-meter tree. We first consider the HMM model as shown in Fig.3. The HMM graph contains t layers (time intervals) and each layer contains 2^n vertices (states). $S_{x_t}^0$ is the vertex reward which means the likelihood that the observation y_t is generated by state x_t . $S_{x_t, x_{t-1}}$ is the edge reward, which indicates the state transition probability. A *path* is a sequence of vertices x_1, x_2, \dots, x_t crossing t layers, whose *reward* is evaluated by the

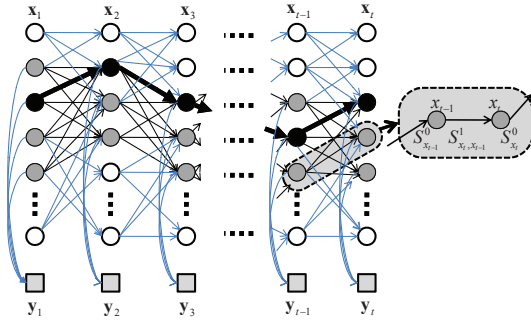


Figure 3: Online Fast Sequence Decoding Algorithm

product of rewards of edges and vertices associated to the path:

$$w(x_1, x_2, \dots, x_t) = \alpha_t \prod_{u=1}^t S_{x_u}^0 \prod_{u=2}^t S_{x_u, x_{u-1}}^1 \quad (8)$$

In (8),

$$\alpha_t = \frac{1}{\sum_{x_1, x_2, \dots, x_t \in \mathbb{P}_t} \prod_{u=1}^t S_{x_u}^0 \prod_{u=2}^t S_{x_u, x_{u-1}}^1}, \quad (9)$$

is a normalizer that keeps the total rewards of all possible paths at time t (denoted by set \mathbb{P}_t) equal to 1. We define $\gamma(x_t)$ as the maximal reward associated to any path from nodes in layer 1 to x_t :

$$\gamma(x_t) = \max_{x_1, x_2, \dots, x_{t-1}} w(x_1, x_2, \dots, x_{t-1}, x_t) \quad (10)$$

Thus, we can calculate the reward of the best path $\gamma(x_t)$ via

$$\gamma(x_t) = \begin{cases} \alpha_1 S_{x_1}^0 & \text{if } t = 1, \\ \max_{x_{t-1}} (\alpha_t \cdot \gamma(x_{t-1}) \cdot S_{x_t, x_{t-1}}^1 S_{x_t}^0) & \text{otherwise.} \end{cases} \quad (11)$$

The sequence decoding algorithm is to find the best reward path from layer 1 to layer t .

$$\gamma(x_t)^* = \max_{x_t} \gamma(x_t) \quad (12)$$

Traditional Viterbi algorithm opens 2^n states at time t to evaluate $S_{x_t}^0$ and backtracks $t - 1$ steps for calculating (11). In each backtrack step, up to 2^n predecessors are opened. We use *open* to mean an operation of reward calculation, so Viterbi has $O(t2^{2n})$ complexity. In this paper, a polynomial time online decoding algorithm is proposed. The main idea is to open only necessary states in the forward and backward steps.

4. Fast Forward Search Strategy

In the forward step at time t , only the states that satisfy constraint (13) will be open. We call them the *feasible states*.

$$\|x_t \mathbf{p} - y_t\|_2 \leq \varepsilon \quad (13)$$

Since in the off-line phase, the 2^n states are ordered according to their energy consumption values, using $y_t - \varepsilon$ and $y_t + \varepsilon$ as search target, we can conduct twice binary searches on the 2^n states, which will find all the feasible states that satisfy (13). ε is the foreknowledge about the metering error bound. Such a binary search step on 2^n states only has complexity $O(\log(2^n)) = O(n)$, which makes the forward search very quick. The vertex reward, i.e., the likelihood that state x_t generates the observation can be calculated by:

$$S_{x_t}^0 = \beta_t \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\|x_t \mathbf{p} - y_t\|_2)^2}{2\sigma^2}} \quad (14)$$

It uses the normal distribution $N(0, \sigma^2)$ as an example, which can be extended to other distributions. σ can be set to ε/k where $k > 3$ for guaranteeing most of metering errors are less than ε . The feasible states at time t are denoted by set \mathbf{F}_t . β_t normalizes the total likelihoods of all feasible states at time t equal to one.

5. Fast Backward Search Strategy

After getting a feasible state x_t by the forward search, a backtrack algorithm is needed to calculate the best path reward $\gamma(x_t)$ by (11). By the state transition model \mathbf{A}_t , the link reward from x_t to a predecessor x_{t-1} can be calculated by:

$$S_{x_t, x_{t-1}}^1 = p^d (1 - p)^{n-d} \quad (15)$$

where $d = \|x_t - x_{t-1}\|_1$ is the number of different states between x_t and x_{t-1} . Since the on/off switching events from $t - 1$ to t is sparse, which is upper bounded by U_t , it is not necessary to open all the predecessor states. At most $\sum_{i=1}^{U_t} \binom{n}{i}$ predecessors

maybe open, which is $O(n^{U_t})$. Further, the infeasible predecessors should not appear in the state sequence, which is needless to

open. So that $\forall x_t \in \mathbf{F}_t$, at most $\min \left\{ \sum_{i=1}^{U_t} \binom{n}{i}, |\mathbf{F}_{t-1}| \right\}$ predecessors

need to be visited for calculating $\gamma(x_t)$, in which $|\mathbf{F}_{t-1}|$ is generally a very small value, guaranteeing the calculation to be very efficient. Since $\gamma(x_t)$ can be fully determined by the feasible predecessors by backtracking only one step without the needs to backtrack t steps. So the backward search algorithm has the worst complexity $O(n^{U_t})$ which is polynomial to n . $\gamma(x_t)^*$ can be calculated by (12). For reliable decoding against the ambiguities, FSD keeps the top- K possible pathes without assertively choosing one top path. The storage cost is linear to t , which is very small.

The forward search and the backward search algorithm can be executed in m mono-meter trees in parallel. The overall online algorithm needs at most $O(n^{U_t+1})$ calculations to calculate the top- K pathes in time t . Since $n < N$, the online algorithm is efficient in tracking real-time states of massive electrical appliances. The normalizer $\{\alpha_t\}$ and $\{\beta_t\}$ can be calculated in real-time, avoiding the difficulties of assigning reward weights.

4. METER DEPLOYMENT OPTIMIZATION

The FSD algorithm gives fast state sequence decoding based on the real-time measurements of smart meters. However, the deployment scenarios of the smart meters will not only dominate the system deployment, maintenance and data collection costs, but also affect the state sequence decoding difficulty and accuracy. Users generally hope to place the least number of meters to get enough information for decoding the on/off sequence of the appliances. Therefore, this section studies the meter deployment optimization problem.

1. Problem Definition

The Meter Deployment Optimization Problem[MDOP] can be defined as follows:

PROBLEM 2 (MDOP PROBLEM). Given a load tree $T = (V, E)$ with N nodes, let $L \subseteq V$ be the set of leaves in T . Each leaf $l_i \in L$ has a power pattern \mathcal{P}_i on it. A subtree $ST(v) = (V(v), E(v))$ denotes the subtree of T with node $v \in V$ as its root. $V(v)$ and $E(v)$ denotes the set of nodes and edges in the subtree $ST(v)$ correspondingly. A binary $x_{i,t} \in \{0, 1\}$ is assigned to each leaf $l_i \in L$, indicating the on/off state of the appliance at time t . If a smart meter is deployed at node v , it can measure the total power

consumed by its subtree $ST(v)$, i.e., measure $\sum_{i \in ST(v)} x_{i,t} \mathcal{P}_i$. The goal of smart meter deployment optimization is to minimize the number of deployed meters while still getting enough information to know the value of each $x_{i,t}$.

2. The Entropy of Meter Measurement

Understanding how the deployment scenario will affect the state tracking accuracy is a fundamental problem in the proposed lightweight energy auditing method. We exploit information entropy to answer this problem. In information theory, entropy (or Shannon's entropy) is a measurement of the uncertainty associated with a random variable. In our problem, sum function of the energy consumptions could be treated as a way to compress the binary state information x_i , which can be thought as a random variable. The goal of the problem is to decode $x_i, i \in ST(v)$ without error, thus the compression should be lossless. From information theory we know that if a compression scheme is lossless, i.e., if we can always recover the entire original message without error, then the compressed message has the same total entropy as the original. That requires all the subtrees which are measuring by a smart meter are lossless. In other words, for each subtree $ST(v)$, the entropy of the sum on the root should be equal to the entropy of the random variables $x_i, i \in ST(v)$ on the leaves. Suppose $Prob(x_i = 0) = q_i$ for leaf i in mono-meter tree $ST(v)$. Then the sum entropy of the leaf nodes can be calculated as:

$$H_s(v) = - \sum_{i \in ST(v)} q_i \log q_i \quad (16)$$

For the smart meter at the root, it can measure only $t \leq 2^n$ distinct aggregated power values, which are corresponding to 2^n appliance states. Consider the i th distinct aggregated power values $a_i \in [a_1, a_2 \dots a_t]$, it may correspond to m_i combinations of states. Let set S_i be any subset of appliances, $i \in \{1, 2, \dots, t\}$ such that $\sum_{j \in S_i} \mathcal{P}_j = a_i$. Since all the appliances take measurements in i.i.d., the probability that the meter measures a value a_i is:

$$Prob(d_v = a_i) = \sum_{S_i} \prod_{j \in S_i, k \in S \setminus S_i} (1 - q_j) q_k \quad (17)$$

in which d_v indicates a measurement at root v . So that the entropy of the smart meter is:

$$H_d(v) = - \sum_{i=1}^t Prob(d_v = a_i) \log(Prob(d_v = a_i)) \quad (18)$$

3. The Clear Ratio

Note that $H_d(v) < H_s(v)$, if there are two different states have the same sum power consumption. That means it will be difficult to disambiguate the exact state from the aggregated power measurement. In this situation, we call the mono-meter tree $ST(v)$ is *Blurry*. Otherwise, if $H_d(v) = H_s(v)$, we call $ST(v)$ is *Clear*, which means the states of appliances can be disaggregated without error.

From information theory, we know that the less information we get, the hard to recover the original state vector. Thus we define the clear ratio $r(D, T)$ for a deployment D on load tree T as:

$$r(D, T) = \min_{v \in D} \frac{H_d(v)}{H_s(v)} \quad (19)$$

The state combinations of appliances can be disaggregated without error when $r(D, T) = 1$ and decoding ambiguities increase as $r(D, T)$ decrease. Given a constant factor r as the threshold of the clear ratio to guarantee the decoding performance, the deployment

optimization problem is to find an optimal deployment D over T to maximize $r(D, T)$ while minimizing the number of deployed meters.

4. MDOP Algorithm for the Bounded Trees

We can prove that this MDOP problem is NP-complete by an polynomial time reduction from the 3-SAT problem. Details will be omitted for the space limitation. Since the MDOP problem is NP-Complete, finding an efficient algorithm that output optimal solution is hard. But in practice, the degree of the power load tree are usually small, and the total power consumptions of the tree are also bounded. Thus we can still design efficient MDOP algorithm for solving practical problems. We make two following assumptions to bound the degree and the maximum power consumption of the tree:

1. the maximum degree of the node in T is upper bounded by a constant d .
2. $\sum_{i \in V} \mathcal{P}_i \leq P_{max}$, P_{max} is a constant.

Then we will introduce an polynomial algorithm for minimizing the number of meter deployment when a clear ratio $r(D, T)$ is given. The algorithm searches tree T node by node from bottom up to the root. Thus all the nodes in subtree $ST(v)$ (except for v) must have been visited before v . Let $T_i, i = 1, 2, \dots, N$ be the remained tree after i^{th} iteration of the algorithm and $ST_i(v)$ be the subtree of T_i rooted on v . Suppose in i^{th} iteration we visit node v . Then we decide meter deployment in $ST_i(v)$ and cut T_i accordingly by checking the clear ratio of the subtree $ST_i(v)$ against the requirement.

1. **If the clear ratio of $ST_i(v)$ is less than $r(D, T)$,** we need to deploy more meters in $ST_i(v)$. Suppose $Children(v)$ is the set of all the children of node v . There are at most 2^d subsets of $Children(v)$. It takes time $O(1)$ to enumerate all of them. Then we could find the smallest subset $C_{best} \subseteq Children(v)$ such that the clear ratio of the subtree that $ST_i(v)$ is less or equal than r by removing all the subtrees $ST_i(u), u \in C_{best}$. Meters should be deployed on node u for all $u \in C_{best}$. And the new remaining tree T_{i+1} is generated from T_i by removing all the subtrees $ST(u), u \in C_{best}$.
2. **If the clear ratio of $ST_i(v)$ is larger or equal than $r(D, T)$,** we need not deploy more meters in $ST_i(v)$. We just connect all leaves of $ST_i(v)$ directly to node v .

We keep this strategy from the bottom up to the root of tree T . Notice that a meter is needed to be deployed on the root if the remaining tree is not empty. The algorithm outputs an deployment strategy D which is at most 2 times the size of an optimal solution for any given T and requirement factor r . And total running time is $O(2^d \mathcal{P}_{max} n^2)$, which is polynomial to n when d and \mathcal{P}_{max} is constant.

The following pseudo-code 1 generally describes algorithm. Notice that we only need T_{i-1} in the i^{th} iteration. The notation of T_i will make the analysis clear, while in practice we only keep the newest tree T in each iteration for the algorithm.

5. NUMERICAL EVALUATIONS

We conduct extensive experiments to evaluate the proposed MDOP algorithm and the FSD algorithm by both the simulated data and the real data from Powernet data set. In simulations, load trees contain N leaf nodes with the maximum D degree were generated randomly, simulating the arbitrary power distribution networks. The power patterns of the electrical appliances (leaf nodes) were generated by Uniform, Normal or Exponential distributions, in which

Algorithm 1 Algorithm for Bounded Tree

INPUT : Tree $T = (V, E)$ with integers on the leaves
OUTPUT: A set $V' \subseteq V$ represent the nodes need to be placed meters.
 $V' = \emptyset$
for all $i \leftarrow 1$ to n **do**
 let v be the i^{th} node when searching from bottom to the root
 if $\text{Clear-Ratio}(ST(v)) < r$ **then**
 Let set $\text{Children}(v) = \{u | u \text{ is a child of } v\}$
 $C_{best} = \emptyset$
 for all $C \subseteq \text{Children}(v)$ **do**
 if $|C_{best}| < |C|$ and $\text{Clear-Ratio}(ST(v) \setminus ST(u), \forall u \in \text{Children}(v) \setminus C) \geq r$ **then**
 $C_{best} \leftarrow C$
 end if
 end for
 $V' = V' \cup \text{Children}(v) \setminus C_{best}$
 $T \leftarrow T \setminus ST(u), \forall u \in \text{Children}(v) \setminus C_{best}$
 end if
end for
OUTPUT: V'

the uniform power distribution simulates the case when appliances' power levels are almost even; normal distribution simulates the general case and exponential distribution simulates the case when appliances' powers are very concentrated.

1. Performance of MDOP

For evaluating MDOP, we evaluate 1) the deployment cost saving performance; 2) the sub-tree character after meter deployment and 3) the effect of clear ratio to the deployment cost.

1. Cost Saving Ratio

We apply MDOP to load trees with 100-1000 electrical appliances and evaluate the *cost saving ratio*, which is defined by the number of deployed smart meters given by MDOP divided by the number of appliances in the load tree. In these experiments, the clear ratio in the MDOP algorithm is set to 1; the normal, uniform and exponential power distributions are evaluated with equal mean = 100. The cost saving results for different load trees are plotted in Fig.4. From the results we see that for load trees with different size and

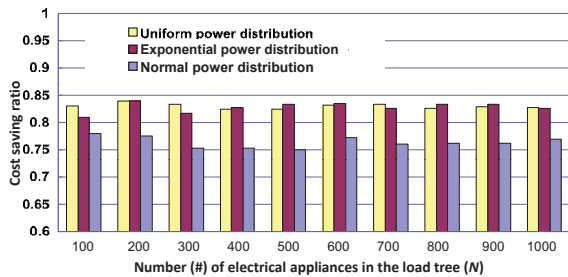


Figure 4: Cost saving ratio of MDOP for different scale load trees and different power pattern distributions

different power distributions, the MDOP algorithm can reduce the meter deployment cost by more than 75% if comparing with the one-to-one monitoring method. The cost saving ratios in exponential and uniform power distributions are similar. When the power levels of appliances are more concentrated (in the normal distribution), more smart meters are required for disambiguating the states of similar-power appliances, which accords our general intuition.

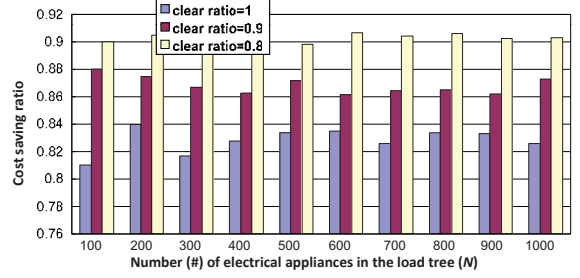


Figure 5: Cost saving ratio VS. Clear ratio

2. Cost Saving Ratio VS. Clear Ratio

Fig.5 further shows how the clear ratio in MDOP algorithm affects the cost saving ratio. The results are shown for load trees with normally distributed power patterns. It can be seen that the cost saving ratio decreases with the clear ratio. When the clear ratio is set to 0.8, almost 90% deployment cost can be saved. The results hold for different power distributions. In next section, we will show the FSD algorithm provides accurate state tracking even when the clear ratio is not high.

3. Size Distribution of the Mono-meter Trees

After meter deployment, the load tree will be split into m mono-meter trees. The size of each mono-meter tree (number of appliances in the tree) dominates the complexity and accuracy of on-line decoding algorithms. We investigate the distribution of mono-meter tree size for different clear ratios. For a load tree with 500 appliances with power distribution $N(200, (200/3)^2)$, we run 20 MDOP experiments for different clear ratios and plot the fitted distribution of the mono-meter tree size in Fig.6. The results show that MDOP generally divide load tree to similar size mono-meter trees. For $r = 1$, most of mono-meter trees have less than 10 leaf nodes (appliances), which guarantees the accurate on/off state disambiguation. When $r = 0.8$ the size of mono-meter trees increase a little; for $r = 0.6$, the size of mono-meter trees is around 15. Since the decoding complexity and ambiguity will increase with the size of mono-meter tree, the results visually show that MDOP with less clear ratios can save more deployment costs but poses higher pressure to the online decoding algorithms.

2. Performance of FSD

The mono-meter trees are generally not large in size, which reduces the online decoding complexity remarkably, i.e., the state space is reduced from 2^N to 2^n , where $n < N$. Therefore, in evaluating FSD, we focus more on the decoding accuracy than the decoding efficiency. The FSD algorithm is run in different mono-meter

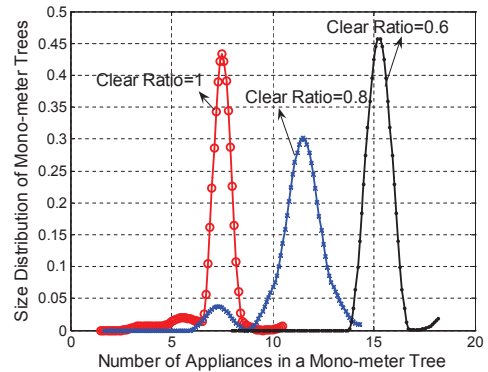


Figure 6: Mono-meter tree size distribution in different clear ratios

trees in parallel. We evaluate the accuracy of FSD over the whole load tree by considering the average accuracy of all the mono-meter trees:

$$e = \frac{1}{T \cdot N} \sum_{t=1}^T \|\mathbf{X}(t) - \hat{\mathbf{X}}(t)\|_1 \quad (20)$$

In (20), $\mathbf{X}(t)$ indicates the ground truth states of N appliances at time t . $\hat{\mathbf{X}}(t)$ is the estimated state vector. e evaluates the average state tracking error over N appliances and over time T . The state tracking error at time t is evaluated by $e(t) = \frac{1}{N} \|\mathbf{X}(t) - \hat{\mathbf{X}}(t)\|_1$.

1. Metering Noise V.S. Tracking Accuracy

We first evaluate how the metering noises of the smart meters affect the tracking accuracy. In simulations, we set the meter noise to $N(0, \sigma^2)$, where $\sigma = \varepsilon/5$ to guarantee most of the metering errors are less than $|\varepsilon|$. For a load tree of 500 nodes with power distribution $N(200, (200/3)^2)$, the real-time tracking errors vs. metering noises σ are plotted in Fig.7. In the experiments, meters are deployed by MDOP with clear ratio $r = 1$; each point is the average result of 10 experiments. The blue curves show the tracking error of FSD and the red curves show the tracking error given by traditional Viterbi algorithm. In Viterbi, the vertex reward is assigned a weight to make it comparable to the link reward. The results show that 1). the FSD algorithm generally has better tracking accuracy than Viterbi. 2) When the metering error is small, the tracking error of FSD is very small, showing its effectiveness in disambiguating the mixed states. The FSD algorithm performs better than Viterbi for it keeping top-500 feasible pathes instead of only the top-path. Another reason is that FSD uses product-type reward function, which doesn't suffer the weight assignment error for balancing the link rewards and the vertex rewards.

2. Clear ratio VS. Tracking accuracy V.S. Cost saving ratio

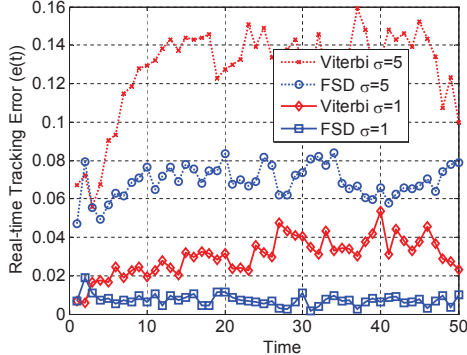


Figure 7: Metering Noise VS. Tracking Accuracy

Then we investigate how the clear ratio of the MDOP algorithm affects the state tracking accuracy. For focusing on the effects of clear ratios, the metering noise σ is set to zero, so all the errors are caused by the decoding ambiguities. For the same load tree settings in Fig.7, the tracking accuracy vs. clear ratio and corresponding cost saving ratio are plotted in Fig. 8. The results show interesting features of this lightweight metering problem. The cost saving ratio increases slowly with the reduction of the clear ratio, and reaches a saturated status when the clear ratio is less than 0.6. The tracking errors increase very quickly with the clear ratio. The different trends of the curves indicate some good region for choosing the clear ratio, in which the tracking error is small and most deployment costs can be saved, as the region 0.8 to 1 in the figure.



Figure 8: Tracking Accuracy vs. Clear ratio vs. Cost Saving Ratio

3. Experiments on Powernet Data Set

Above experiments assume appliances have static power patterns. We conduct further experiments using the Powernet data to relax this assumption. We use the data of Sep-30-2011 which contains feasible data of 65 appliances. Some appliances are not open on that day and some appliances have very small power are not used. We firstly evaluate the power patterns of these appliances. By statistical analysis on 500 samples for each appliance, the Std/Mean for each appliance is evaluated, which is plot in Fig.9. It can be seen that more than 75% appliances have Std/Mean less than 0.1, indicating the power consumption of appliances in real applications are not highly dynamic.

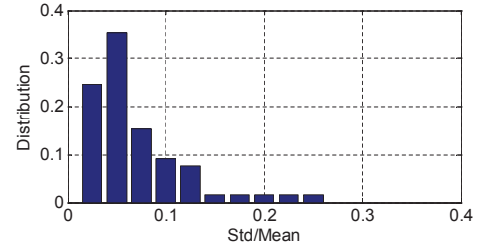


Figure 9: Power Std/Mean for 65 appliances in Powernet

Since the dataset doesn't provide the load tree structure. We off-line train each appliance's power pattern by the average energy consumption over 10 minutes, and randomly generate a load tree to assign 65 appliances randomly to the leaves. Smart meters are deployed by MDOP with $r = 1$ on the generated load tree. In on-line phase, when an appliance is on, its energy consumption is not static but follows its energy consumption trace in the data set. The meters measure the past 30 seconds moving average of the mixed real-time energy consumption of its subtree to decode the states of appliances. The state tracking performances by FSD and Viterbi are plotted in Fig.10. The decoding accuracy is generally around 20% for FSD, which shows potential of the proposed framework considering its current rough model and large cost saving ratio. We checked the errors and found the main reason is because the variation range of some large power alliances in the same mono-meter tree covers the on/off events of the small power appliances. Such problems can be further resolved by improving meter deployment scheme to consider both the mean and the variance of different electrical appliances, which will be studied in future work.

6. CONCLUSION AND FUTURE WORK

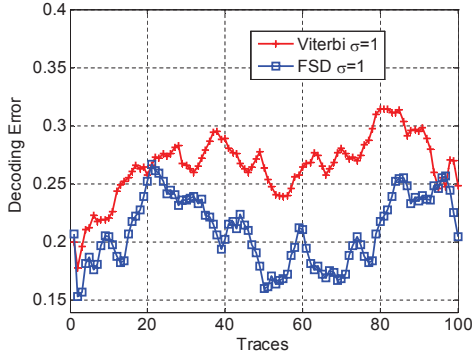


Figure 10: Tracking accuracy using powernet data set

This paper presents a lightweight metering and sequence decoding framework for tracking the on/off states of electrical appliances. The rationale is that the power patterns of appliances can be learned off-line and the switching events of the electrical appliances in a short interval are sparse. FSD, a fast state sequence decoding algorithm is proposed by off-line tree splitting, state vector pre-ordering, online forward search and backward search algorithms. It facilitates a polynomial time decoding algorithm which overcomes the complexities of disambiguating 2^N states. By entropy-based analysis, a “clear ratio” is proposed to bridge the deployment cost and the tracking accuracy, which can be seen as a parameter to describe the decoding accuracy requirement. Based on it, MDOP, a polynomial time deployment algorithm is proposed to deploy the minimal number of smart meters for a given requirement of the clear ratio. The experimental results show the effectiveness and good performances of the proposed methods.

This work contains some basic assumptions, such as the power patterns are static, the state transition probabilities are i.i.d. In future work, more complex power patterns and robust deployment algorithm will be studied. The state transitions can be further modeled by Hidden Semi Markov Model to consider the work duration distributions of the appliances. Detection of transient signals can help to extract the occasion of the switching events, and the group dependence of on/off switching events can further increase the decoding accuracy.

7. REFERENCES

- [1] <http://www.st-hughs.ox.ac.uk/news-and-events/archive/saving-energy-at-st-hughs>.
- [2] Leslie K Norford and Steven 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.
- [3] 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):1200–1210, jul 1995.
- [4] J. Lifton, M. Feldmeier, Y. Ono et al. A platform for ubiquitous sensor deployment in occupational and domestic environments. In *IPSN '07*, pages 119–127, New York, NY, USA, 2007. ACM.
- [5] Xiaofan Jiang, Stephen Dawson-Haggerty, Prabal Dutta, and David Culler. Design and implementation of a high-fidelity ac metering network. In *Proceedings of ACM IPSN*, IPSN '09, pages 253–264, Washington, DC, USA, 2009. IEEE Computer Society.
- [6] Xiaofan Jiang, Minh Van Ly, Jay Taneja, Prabal Dutta, and David Culler. Experiences with a high-fidelity wireless building energy auditing network. In *SenSys '09*, pages 113–126, New York, NY, USA, 2009. ACM.
- [7] S. Dawson-Haggerty, S. Lanzisera, J. Taneja et al. @scale: insights from a Large, long-lived appliance energy WSN. In *Proceedings of ACM IPSN*, Beijing, China, 2012.

- [8] PowerNet, <http://powernet.stanford.edu/index.php>.
- [9] Deokwoo Jung and Andreas Savvides. Estimating building consumption breakdowns using on/off state sensing and incremental sub-meter deployment. In *SenSys '10*, pages 225–238, New York, NY, USA, 2010. ACM.
- [10] The tendril residential energy ecosystem. <http://www.tendrillinc.com/>.
- [11] Green Box. <http://www.getgreenbox.com/>.
- [12] Energy Hub. <http://www.energyhub.net/>.
- [13] Sensicast. <http://www.sensicast.com/>.
- [14] G. W. Hart. Nonintrusive appliance load monitoring. In *Proceedings of the IEEE*, pages 1870–1891, December 1992.
- [15] Shwetak N Patel, Thomas Robertson, Julie A Kientz, Matthew S Reynolds, and Gregory D Abowd. At the flick of a switch : Detecting and classifying unique electrical events on the residential power line. In *Ubicomp '07*, pages 271–288, 2007.
- [16] Linda Farinaccio and Radu Zmeureanu. Using a pattern recognition approach to disaggregate the total electricity consumption in a house into the major end-uses. *Energy and Buildings*, 30(3):245–259, 1999.
- [17] Y. Kim, T. Schmid, Z. Charbiwala et al. Viridiscopes: design and implementation of a fine grained power monitoring system for homes. In *Ubicomp '09*, pages 245–254, New York, NY, USA, 2009. ACM.
- [18] Sidhant Gupta, Matthew S. Reynolds, and Shwetak N. Patel. Electrisense: single-point sensing using emi for electrical event detection and classification in the home. In *Ubicomp '10*, pages 139–148, New York, NY, USA, 2010. ACM.
- [19] Anthony Rowe, Mario Berges, and Raj Rajkumar. Contactless sensing of appliance state transitions through variations in electromagnetic fields. In *BuildSys '10*, pages 19–24, New York, NY, USA, 2010. ACM.
- [20] Z. Cihan Taysi, M. Amac Guvensan, and Tommaso Melodia. Tinyyears: spying on house appliances with audio sensor nodes. In *BuildSys '10*, pages 31–36, New York, NY, USA, 2010. ACM.
- [21] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.
- [22] http://en.wikipedia.org/wiki/Viterbi_algorithm.

APPENDIX

In this section, we prove that the objective function in (2), can be transformed approximately to the objective function in (3). Mathematically,

$$\begin{aligned} & \mathbb{E}\{(\mathbf{Z}_{t,i} - \sum_{j \in S(i)} \mathbf{X}_{t,j} \mathbf{P}_{t,j})^2\} \\ &= \mathbf{Z}_{t,i}^2 - 2\mathbf{Z}_{t,i} \sum_{j \in S(i)} \mathbf{X}_{t,j} \mathbf{P}_{t,j} + \mathbb{E}\{\sum_{j \in S(i)} \mathbf{X}_{t,j} \mathbf{P}_{t,j}\}^2 \quad (21) \\ &= (\mathbf{Z}_{t,i} - \sum_{j \in S(i)} \mathbf{X}_{t,j} \mathbf{P}_{t,j})^2 + \Delta, \end{aligned}$$

where

$$\begin{aligned} \Delta &= \mathbb{E}(\sum_{j \in S(i)} \mathbf{X}_{t,j} \mathbf{P}_{t,j})^2 - (\sum_{j \in S(i)} \mathbf{X}_{t,j} \mathbf{P}_{t,j})^2 \\ &= \sum_{j \in S(i)} \mathbf{X}_{t,j}^2 \delta_j. \quad (22) \end{aligned}$$

Note that $\mathbf{X}_{t,j}^2 = \mathbf{X}_{t,j}$, and $\delta_j \approx \alpha \mathbf{P}_{t,j}$, we have

$$\begin{aligned} & \mathbb{E}\{(\mathbf{Z}_{t,i} - \sum_{j \in S(i)} \mathbf{X}_{t,j} \mathbf{P}_{t,j})^2\} \\ &= \mathbf{Z}_{t,i}^2 - (2\mathbf{Z}_{t,i} - \alpha) \sum_{j \in S(i)} \mathbf{X}_{t,j} \mathbf{P}_{t,j} \\ & \quad + (\sum_{j \in S(i)} \mathbf{X}_{t,j} \mathbf{P}_{t,j})^2 \quad (23) \\ &= (\mathbf{Z}_{t,i} - \alpha/2 - \sum_{j \in S(i)} \mathbf{X}_{t,j} \mathbf{P}_{t,j})^2 + \Lambda, \end{aligned}$$

where $\Lambda = \alpha \mathbf{Z}_{t,i} - \alpha^2/4$, is a constant.