Energy Disaggregation via Adaptive Filtering

Roy Dong, Lillian J. Ratliff, Henrik Ohlsson, and S. Shankar Sastry

Abstract— **The energy disaggregation problem is recovering device level power consumption signals from the aggregate power consumption signal for a building. We show in this paper how the disaggregation problem can be reformulated as an adaptive filtering problem. This gives both a novel disaggregation algorithm and a better theoretical understanding for disaggregation. In particular, we show how the disaggregation problem can be solved online using a filter bank and discuss its optimality.**

I. INTRODUCTION

Power consumption data of individual devices have the potential to greatly decrease costs in the electricity grid. Currently, residential and commercial buildings account for 40% of total energy consumption [1], and studies have estimated that 20% of this consumption could be avoided with efficiency improvements with little to no cost [2], [3]. It is believed that the largest barrier to achieving these energy cost reductions is due to behavioral reasons [4]. However, smart meter technology currently only provides the aggregate power consumption data, and deployment is at a sufficiently advanced stage that a change in hardware is prohibitively expensive.

Disaggregation presents a way in which device-level consumption patterns of individuals can be learned by the utility company. This information would allow the utility to present this information to the consumer, with the goal of increasing consumer awareness about energy usage. Studies have shown that this is sufficient to improve consumption patterns [5].

Our aim in this paper is to formulate the disaggregation problem in the filter banks framework. In particular, we develop an algorithm for disaggregation of whole building energy data using dynamical models for devices and filter banks for determining the most likely inputs to the dynamical models. Under mild assumptions on the noise characteristics we are able to provide guarantees for when the algorithm recovers the disaggregated signal that most closely matches our observed data and priors.

In Section II, we discuss previous work on the topic of energy disaggregation. In Section III, we formally define the problem of energy disaggregation. In Sections IV-A to IV-B, we establish our framework for solving the problem of energy disaggregation. In Section V, we provide an online adaptive filtering algorithm for estimating individual device power consumption patterns and quickly summarize theoretical results. Finally, in Section VI, we give concluding remarks.

II. BACKGROUND

The problem of energy disaggregation, and the existing hardware for disaggregation, has been studied extensively in the literature (see [6], [7], for example). The goal of the current disaggregation literature is to present methods for improving energy monitoring at the consumer level without having to place sensors at device level, but rather use existing sensors at the whole building level. The concept of disaggregation is not new; however, only recently has it gained attention in the energy research domain.

Disaggregation, in essence, is a single-channel source separation problem. The problem of recovering the components of an aggregate signal is an inverse problem and as such is, in general, ill-posed. Most disaggregation algorithms are batch algorithms and produce an estimate of the disaggregated signals given a batch of aggregate recordings. There have been a number of survey papers summarizing the existing methods (e.g. see [8], [9]). In an effort to be as self-contained as possible, we try to provide a broad overview of the existing methods and then explain how the disaggregation method presented in this paper differs from existing solutions.

The literature can be divided into two main approaches, namely, supervised and unsupervised. Supervised disaggregation methods require a disaggregated data set for training. This data set could be obtained by, for example, monitoring typical appliances using plug sensors. Supervised methods assume that the variations between signatures for the same type of appliances is less than that between signatures of different types of appliances. Hence, the disaggregated data set does not need to be from the building that the supervised algorithm is designed for. However, the disaggregated data set must be collected prior to deployment, and come from appliances of a similar type to those in the target building.

Unsupervised methods, on the other hand, do not require a disaggregated data set to be collected. They do, however, require hand tuning of parameters, which can make it hard for the methods to be generalized in practice. It should be said that also supervised methods have tuning parameters, but these can often be tuned using the training data.

The existing supervised methods include sparse coding [10], change detection and clustering based approaches [11], [12] and pattern recognition [13]. The sparse coding approach tries to reconstruct the aggregate signal by selecting as few signatures as possible from a library of typical signatures. Similarly, in our proposed framework we construct

R. Dong, L. Ratliff, H. Ohlsson, and S. S. Sastry are with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, CA, USA {roydong, ratliffl, ohlsson, sastry} @eecs.berkeley.edu

H. Ohlsson is also with the Division of Automatic Control, Department of Electrical Engineering, Linköping University, Sweden.

a library of dynamical models and reconstruct the aggregate signal by using as few as possible of these models.

The existing unsupervised methods include factorial hidden Markov models (HMMs), difference hidden Markov models and variants [14], [15], [16], [17], [18] and temporal motif mining [19]. These approaches usually assign a constant power consumption level to a Markov state, and encode the device usage and signatures in the transition probabilities. Our framework is similar to HMM frameworks in the sense that both methods essentially formulate hypotheses on which devices are on at each time instant. However, in contrast to HMMs, in the filter bank framework we incorporate the use of dynamical models to explicitly capture the transients of the devices, which helps identify them.

In this paper, we model devices as linear systems. The novelty of this framework over some other supervised methods is that we explicitly model how the devices are used (with an input prior) and the device signatures (with a device model) separately. More specifically, we formulate hypotheses on the on/off state of the devices over the time horizon for which we have data. The on/off state corresponds to whether the input is activated or not. Using filter banks and the dynamical models we have for device behavior, we evaluate which is the most likely hypothesis on the inputs. We provide an algorithm for this process. Under mild assumptions on the noise characteristics we are able to provide guarantees for when the our algorithm results in an optimal solution.

III. PROBLEM FORMULATION

In this section, we formalize the problem of energy disaggregation.

Suppose we are given an aggregated power consumption signal for a building. We denote this data as $y[t]$ for $t = 0, 1, \dots, T$, where $y[t]$ is the aggregate power consumption at time t. The entire signal will be referred to as y. This signal is the aggregate of the power consumption signal of several individual devices:

$$y[t] = \sum_{i=1}^{D} y_i[t] \text{ for } t = 0, 1, \dots, T, \tag{1}$$

where D is the number of devices in the building and $y_i[t]$ is the power consumption of device i at time t. The goal of disaggregation is to recover y_i for $i = 1, 2, \dots, D$ from y.

To solve this problem, it is necessary to impose additional assumptions on the signals y_i and the number of devices D.

IV. PROPOSED FRAMEWORK

At a high level, our framework can be summarized as follows. First, in the training phase of our disaggregation framework, we assume we have access to a training set of individual device power consumption data that is representative of the devices in the buildings of concern. From this training data, we build a library of models for individual devices. With these models, the disaggregation step becomes finding the most likely inputs to these devices that produces our observed output, the aggregate power consumption signal.

A. Training phase

Suppose we have a training data set, which consists of the power consumption signals of individual devices. Let $z_i[t]$ for $t = 0, 1, \dots, T_i$ be a power consumption signal for a device i. Then, $\{z_i\}_{i=1}^{D}$ is our training data. From this training data, we will learn models for individual devices.

For device i, we assume the dynamics take the form of a finite impulse response (FIR) model:

$$z_i[t] = \sum_{j=0}^{n_i} b_{i,j} u_i^z[t-j] + e_i[t], \tag{2}$$

where n_i is the order of the FIR model corresponding to device i, $b_{i,j}$ represent the parameters of the FIR model and $e_i[t]$ is white noise, i.e. random variables that are zero mean, finite variance, and independent across both time and devices. Furthermore, $u_i^z[t]$ represents the input to device i at time t in the training dataset, z.

We assume FIR models fed by piecewise constant inputs give a rich enough setup to model the energy consumed by individual appliances. Note that many electrical appliances can be seen having a piecewise constant input. For example, the input of a conventional oven can be seen as 0°F if the oven is off, and 300°F if the oven is set to heat to 300°F. Note that the input is not the actual internal temperature of the oven, but rather the temperature setting on the oven. Since the temperature setting is relatively infrequently changed, the input is piecewise constant over time. Many other appliances are either on or off, for example lights, and can be seen having a binary input with infrequent changes. This is also a piecewise constant input. For a washing machine, we have a discrete change between modes (washing, spinning, etc.) and this mode sequence can be seen as the piecewise constant input of the washing machine.

In most applications, we will not have access to any input data. Thus, our system identification step becomes estimation of both the input and the FIR parameters. This is known as a blind system identification problem, and is generally very difficult. However, with the assumption that the inputs represent an on/off sequence, we can use simple change detection methods to estimate the binary input u_i^z.

Additionaly, although n_i is not known a priori, we can select the value of n_i using criterion from the system identification and statistics literature.

Finally, we can succinctly rewrite (2) in vector form:

$$z_i[t] = \boldsymbol{\beta}_i^\top \boldsymbol{\xi}_i[t] + e_i[t], \tag{3}$$

where $\boldsymbol{\beta}_i = \begin{bmatrix} b_{i,0} & b_{i,1} & \dots & b_{i,n_i} \end{bmatrix}^\top$ are the FIR parameters and $\boldsymbol{\xi}_i[t] = \begin{bmatrix} u_i^z[t] & u_i^z[t-1] & \dots & u_i^z[t-n_i] \end{bmatrix}^\top$ are the regressors at time t.

B. Energy disaggregation

Suppose we have estimated a library of models for devices $i = 1, 2, \dots, D$. That is, we are given $\boldsymbol{\beta}_i$ for devices $i = 1, 2, \dots, D$. Furthermore, we are given y. We wish to find y_i for $i = 1, 2, \dots, D$. Now we assume that all devices are modeled in our library, or, equivalently, all devices are represented in our training data. This is a common assumption in the disaggregation literature but we plan to relax this assumption in future work.

Now, this problem is equivalent to finding inputs to our devices that generate our observed aggregated signal. More explicitly, let $\boldsymbol{\beta} = \begin{bmatrix}\boldsymbol{\beta}_1^\top & \boldsymbol{\beta}_2^\top & \dots & \boldsymbol{\beta}_D^\top\end{bmatrix}^\top$ and $\boldsymbol{\psi}[t] = \begin{bmatrix}\boldsymbol{\psi}_1[t]^\top & \boldsymbol{\psi}_2[t]^\top & \dots & \boldsymbol{\psi}_D[t]^\top\end{bmatrix}^\top$, where $\boldsymbol{\beta}_i$ are as defined in Section IV for each device $i = 1, 2, \dots, D$ and $\boldsymbol{\psi}_i[t] = \begin{bmatrix}u_i[t] & u_i[t-1] & \dots & u_i[t-n_i]\end{bmatrix}^\top$. Then, we have a model for the aggregate power signal:

$$y[t] = \boldsymbol{\beta}^\top \boldsymbol{\psi}[t] + e[t], \tag{4}$$

where $e[t] = \sum_{i=1}^{D} e_i[t]$ is still white noise. For simplicity, we assume zero initial conditions, i.e. $u_i[t] = 0$ for $t = -n_i, -n_i+1, \dots, -1$. This assumption can easily be relaxed. Thus, the problem of energy disaggregation is now finding $u_i[t]$ for $t = 0, 1, \dots, T$ and $i = 1, 2, \dots, D$.

Let $\boldsymbol{u}[t] = \begin{bmatrix}u_1[t] & u_2[t] & \dots & u_D[t]\end{bmatrix}^\top$. Recall that in the training phase we assumed that FIR models fed by piecewise constant inputs gave a rich enough setup for accurately modeling energy consumption of individual appliances. We will in the disaggregation step similarly assume that $u_i[t], i = 1, \dots, D$, are piecewise constant over time. It follows that the vector-valued function $\boldsymbol{u}$ is piecewise constant.

Let a segment be defined as an interval in which $\boldsymbol{u}$ is constant. Then, energy disaggregation becomes a segmentation problem. More formally, let $\delta \in \{0, 1\}^T$ be such that $\delta[t] = 1$ if $\boldsymbol{u}[t] \neq \boldsymbol{u}[t-1]$, and 0 otherwise. In other words, $\delta[t]$ is a binary variable that equals 1 if and only if the input changes between times $t-1$ and t.

Next, suppose we are given a segmentation δ, and let $n = \|\delta\|_1$, the number of segments in δ. Then for each device i, we can define a function $\bar{u}_i : \{1, 2, \dots, n\} \to \mathbb{R}$ such that $\bar{u}_i(l)$ is the value of u in segment l. Then, let $\bar{\boldsymbol{u}} : \{1, 2, \dots, n\} \to \mathbb{R}^D, l \mapsto (\bar{u}_1(l), \bar{u}_2(l), \dots, \bar{u}_D(l))$. $\bar{\boldsymbol{u}}(l)$ represents the input to all devices in the lth segment. Also, let y^t denote all measurements available at time t. That is, $y^t = (y[0], y[1], \dots, y[t])$.

Let $p(\boldsymbol{u})$ denote a probability distribution on the user's input to the devices; that is, $p(\boldsymbol{u})$ is the likelihood of the input $\boldsymbol{u}$. This encapsulates our prior on user consumption patterns. For example, in residential buildings, power consumption tends to be low mid-day, while in commercial buildings, power consumption drops off after work hours. This knowledge can be represented in $p(\boldsymbol{u})$.

The disaggregation problem is to find the maximum a posteriori (MAP) estimate of $\boldsymbol{u}$ and, consequently, the power consumption of device i, given our observations. In Section V, we provide an adaptive filtering algorithm for solving this problem.

V. ENERGY DISAGGREGATION VIA ADAPTIVE FILTERING

A. Algorithm definition

In this section, we provide a tractable algorithm to solve the problem posed in Section IV. Furthermore, this algorithm is defined recursively on measurements across time, so it can be run online.

We draw on results in the adaptive filtering literature. An adaptive filter is any filter that adjusts its own parameters based on observations. In our particular case, we use a filter bank approach to handle the problem presented in Section IV. A filter bank is a collection of filters, and the adaptive element of a filter bank is in the insertion and deletion of filters, as well as the selection of the optimal filter.

We will define a filter bank, and also the problem a filter bank solves. Suppose we are given measurements y^t. We wish to find the maximum a posteriori estimate of the input $\boldsymbol{u}$ given our measurements y^t: we wish to find $\boldsymbol{u}$ that maximizes $p(\boldsymbol{u}|y^t)$, which is equivalent to maximizing $p(y^t|\boldsymbol{u})p(\boldsymbol{u})$. Decomposing u into δ and $\bar{\boldsymbol{u}}$, we can again rewrite this as maximizing $p(y^t|\bar{\boldsymbol{u}}, \delta)p(\bar{\boldsymbol{u}}|\delta)p(\delta)$. Note that we can calculate:

$$p(\delta) = \int p(\bar{\boldsymbol{u}}, \delta) d\bar{\boldsymbol{u}}. \tag{5}$$

The final manipulation is that we wish to find a δ to maximize the following quantity:

$$\max_{\bar{\boldsymbol{u}}} \; p(y^t|\bar{\boldsymbol{u}}, \delta)p(\bar{\boldsymbol{u}}|\delta)p(\delta). \tag{6}$$

Now, our algorithm maintains a collection of filters, known as a filter bank. Let $\mathcal{F}$ denote this filter bank. Each filter $f \in \mathcal{F}$ corresponds to a segmentation $\delta_f \in \{0, 1\}^t$. Given each δ_f, we can calculate:

$$\bar{\boldsymbol{u}}_f = \underset{\bar{\boldsymbol{u}}}{\operatorname{argmax}} \; p(y^t|\bar{\boldsymbol{u}}, \delta_f)p(\bar{\boldsymbol{u}}|\delta_f)p(\delta_f). \tag{7}$$

$$p_f = \max_{\bar{\boldsymbol{u}}} \; p(y^t|\bar{\boldsymbol{u}}, \delta_f)p(\bar{\boldsymbol{u}}|\delta_f)p(\delta_f). \tag{8}$$

There are only finitely many possible δ. Thus, if we kept a filter f for every possible segmentation δ, we could easily find the MAP estimate of $\boldsymbol{u}$. However, the filter bank $\mathcal{F}$ would grow exponentially with time. Thus, we need to find methods to keep the size of $\mathcal{F}$ under control.

The process of finding the best segmentation can be seen as exploring a binary tree. That is, a segmentation δ can be thought of as a leaf node on a binary tree of depth t. This is visualized in Figure 1.

Limiting the growth of $\mathcal{F}$ can be done by deciding which branches to expand and which branches to prune. This sort of formulation lends itself very easily to an online formulation of the filter banks algorithm. In fact, it is more intuitive to think of the algorithm in an online fashion. At time t, we choose to branch a filter only if it corresponds to one of the most likely segmentations. By branching, we refer to exploring both the 0 and 1 branches. This is depicted by the blue and green lines in Figure 1. Otherwise, we will merely extend the last segment of the segmentation, i.e. only follow the 0 branch. Additionally, at time t, we prune any paths that have sufficiently low likelihood. That is, we remove the filter f from $\mathcal{F}$ if $p_f < p_{\text{thres}}$, where p_{thres} is an algorithm parameter. This is depicted by the red dotted line in Figure 1.

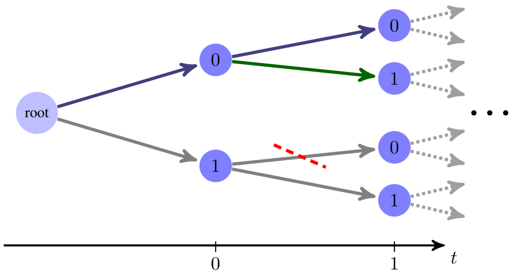

Fig. 1. A segmentation δ can be thought of as a leaf node on a binary tree of depth t. That is, δ corresponds exactly to one leaf node of this binary tree. If we choose to branch the blue path (0) at time 0, then we add the green path $(0, 1)$ as well as the blue path $(0, 0)$. If we choose not to branch (0), then only $(0, 0)$ would be added to our filter bank. The red dotted-line depicts pruning; this would involve removing $(1, 0)$ and all its children from the tree.

Finally, we can exhibit our algorithm. It is presented in Figure 2. In [20], we derived theoretical results guaranteeing that the MAP estimates found by this algorithm are optimal under mild assumptions.

VI. CONCLUSIONS

In the work presented, we formalized the disaggregation problem within the filter banks framework. We provide an algorithm with guarantees on the recovery of the true solution given some assumptions on the data.

REFERENCES

[1] L. Perez-Lombard, J. Ortiz, and C. Pout, “A review on buildings energy consumption information,” *Energy and buildings*, vol. 40, pp. 394–398, 2008.

[2] J. Creyts, A. Derkach, S. Nyquist, K. Ostrowski, and J. Stephenson, “Reducing U.S. greenhouse gas emissions: How much at what cost?” U.S. Greenhouse Gas Abatement Mapping Initiative, Tech. Rep., 2007.

[3] J. A. Ò. Laitner, K. Ehrhardt-Martinez, and V. McKinney, “Examining the scale of the behaviour energy efficiency continuum,” in *European Council for an Energy Efficient Economy*, 2009.

[4] G. Crabtree, “Energy future report: ‘energy future: think efficiency’,” American Physical Society, Tech. Rep., 2008.

[5] K. Ehrhardt-Martinez, K. A. Donnelly, and J. A. Laitner, “Advanced metering initiatives and residential feedback programs: A meta-review for household electricity-saving opportunities,” American Council for an Energy-Efficient Economy, Tech. Rep., June 2010.

[6] M. E. Berges, E. Goldman, H. S. Matthews, and L. Soibelman, “Enhancing electricity audits in residential buildings with nonintrusive load monitoring,” *Journal of Industrial Ecology*, vol. 14, no. 5, pp. 844–858, 2010.

[7] M. Berges, E. Goldman, H. S. Matthews, and L. Soibelman, “Learning systems for electric consumption of buildings,” in *ASCI International Workshop on Computing in Civil Engineering*, 2009.

[8] M. Zeifman and K. Roth, “Nonintrusive appliance load monitoring: Review and outlook,” *IEEE Transactions on Consumer Electronics,*, vol. 57, no. 1, pp. 76–84, Feb. 2011.

[9] J. Z. Kolter and M. J. Johnson, “REDD: A public data set for energy disaggregation research,” in *Proceedings of the SustKDD Workshop on Data Mining Appliations in Sustainbility*, 2011.

[10] J. Z. Kolter and A. Y. Ng, “Energy disaggregation via discriminative sparse coding,” in *Neural Information Processing Systems*, 2010.

[11] S. Drenker and A. Kader, “Nonintrusive monitoring of electric loads,” *IEEE Computer Applications in Power,*, vol. 12, no. 4, pp. 47–51, 1999.

[12] D. Rahayu, B. Narayanaswamy, S. Krishnaswamy, C. Labbé, and D. P. Seetharam, “Learning to be energy-wise: discriminative methods for load disaggregation,” in *Third International Conference on Future Energy Systems: Where Energy, Computing and Communication Meet (e-Energy), 2012*. IEEE, 2012, pp. 1–4.

[13] L. Farinaccio and R. Zmeureanu, “Using a pattern recognition approach to disaggregate the total electricity consumption in a house into the major end-uses,” *Energy and Buildings*, vol. 30, no. 3, pp. 245–259, 1999.

[14] H. Kim, M. Marwah, M. F. Arlitt, G. Lyon, and J. Han, “Unsupervised disaggregation of low frequency power measurements,” in *SDM’11*, 2011, pp. 747–758.

[15] J. Z. Kolter and T. Jaakkola, “Approximate inference in additive factorial HMMs with application to energy disaggregation,” in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2012.

[16] M. J. Johnson and A. S. Willsky, “Bayesian Nonparametric Hidden Semi-Markov Models,” Massachusetts Institute of Technology, Tech. Rep. arXiv:1203.1365, 2012.

[17] O. Parson, S. Ghosh, M. Weal, and A. Rogers, “Nonintrusive load monitoring using prior models of general appliance types,” in *26th AAAI Conference on Artificial Intelligence*, 2012.

[18] S. Pattem, “Unsupervised disaggregation for non-intrusive load monitoring,” in *11th International Conference on Machine Learning and Applications (ICMLA), 2012*, vol. 2. IEEE, 2012, pp. 515–520.

[19] H. Shao, M. Marwah, and N. Ramakrishnan, “A temporal motif mining approach to unsupervised energy disaggregation,” in *Proceedings of 1st International Non-Intrusive Load Monitoring Workshop*, 2012.

[20] R. Dong, L. J. Ratliff, H. Ohlsson, and S. S. Sastry, “Energy disaggregation via adaptive filtering,” in *Proceedings of the 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Oct 2013, pp. 173–180.

```
Initialize $t \leftarrow 0$, $f_0 \leftarrow \delta_{f_0} = (0)$, $f_1 \leftarrow \delta_{f_1} = (1)$, $\mathcal{F} \leftarrow \{f_0, f_1\}$, and $p_{\text{thres}}$.
while TRUE do
    // Find the filters that correspond to the most likely
    // segmentations given $y^t$.
    $\mathcal{F}' \leftarrow \emptyset$.
    for $f \in \mathcal{F}$ do
        if $p_f = \max_{f' \in \mathcal{F}} p_{f'}$ then
            Add a copy of $f$ to $\mathcal{F}'$.
        end if
    end for
    // When available, update all filters in $\mathcal{F}$ with the
    // new measurement.
    Wait for new measurement $y[t+1]$.
    for $f \in \mathcal{F}$ do
        Append 0 to $\delta_f$. Recalculate $\bar{\boldsymbol{u}}_f$ and $p_f$.
    end for
    // Branch the filters corresponding to the most likely
    // segmentations given $y^t$.
    for $f' \in \mathcal{F}'$ do
        Append 1 to $\delta_{f'}$.
        Recalculate $\bar{\boldsymbol{u}}_{f'}$ and $p_{f'}$.
        Add $f'$ to $\mathcal{F}$.
    end for
    // Prune elements from the filter bank that have
    // unlikely segmentations.
    for $f \in \mathcal{F}$ do
        if $p_f < p_{\text{thres}}$. then
            Remove $f$ from $\mathcal{F}$.
        end if
    end for
    $t \leftarrow t + 1$.
end while
```

Fig. 2. Algorithm for online energy disaggregation via filter banks.