2014 IEEE International Conference on Data Mining Workshop

Smart Saver: a Consumer-Oriented Web Service for Energy Disaggregation

Guoming Tang, Jie Chen, Cheng Chen, Kui Wu
Dept. of Computer Science, University of Victoria, B.C., Canada
{guoming, jiechen, cchenv, wkui}@uvic.ca

***Abstract*—Energy disaggregation, which aims to break down the total energy consumption of household into that of individual appliances, plays an important role in energy conservation and has caught more and more attention. Realizing that current energy disaggregation approaches are hard to perform for the ordinary consumers and free and open applications/services are not generally available, we provide a consumer-oriented web service, Smart Saver, which is not only open and free to the consumers but also user-friendly and easy-to-use for energy disaggregation. Based on a simple power model, a sparse switching event recovery model is established as the core of Smart Saver. By feeding the basic power information of appliances into Smart Saver, the users will be provided with 1) online energy disaggregation and appliance monitoring if they have smart meters communicating with our service, or 2) offline energy disaggregation if they upload their aggregated power data.**

I. Introduction

Nowadays, utilities around the world are deploying smart meters to lower the cost of recording and reporting consumers' electricity consumption. The consumers, however, have not found much benefit from the "smart" devices, except a time series of meter readings online. Is there any way we could take advantage of the meter readings? What if we could make use of such data and figure out the energy consumption for each appliance in our houses?

Energy disaggregation, also known as non-intrusive load monitoring (NILM), as illustrated in Fig. 1, aims to learn the energy consumption of individual appliances from their aggregated energy consumption values, *e.g.*, the total energy consumption of a house. With accurate energy disaggregation, the consumers can 1) learn how much energy each appliance consumes, 2) take effective measures to save energy, and 3) play more active roles in demand response (DR) programs.

More and more consumers can get access to their aggregated load data with the help of smart meters. There are, however, still stumbling blocks on the road to perform energy disaggregation by ordinary consumers. Even though a broad spectrum of solutions to energy disaggregation have been proposed since 1980s [3], [11], most are based on appliances' energy usage patterns, also called *signatures* of appliances. Those signatures are hard to obtain without particular machine learning techniques or auxiliary measurements. For example, in [2], extra equipments are needed to detect the activities of appliances based on high frequency electromagnetic interference (EMI). Other solutions like [5], [6], [7], [9] need a lot of training or inference work, which is too complex to handle by ordinary consumers.

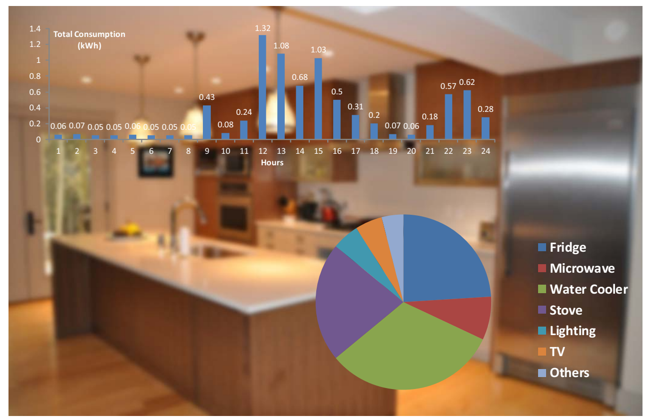

Fig. 1. An example of energy disaggregation for one-day energy consumption of an apartment

Recently, a rapidly growing number of start-ups, such as Bidgely, PlotWatt, Navetas, and Energy Aware, are established to provide commercial services for energy disaggregation in consumers' houses. Those private services, however, are usually not cheap and without guarantee for accuracy. Furthermore, special measuring instruments need to install in consumers' houses. So far, *no* free/open service/application has been found to provide public energy disaggregation for common households without installing any extra equipments.

We have observed that 1) the *power information* of an appliance, such as rated and stand-by power, is normally available in practice, from users' manual, technical specification or public web sites such as [1], and 2) low-cost plug-and-play power meters are popular on current market and in some consumers' houses [4]. In this demo, we challenge the traditional energy disaggregation solutions and try to 1) establish a *simple* and *universal* model for energy disaggregation only referring to readily-available information of appliances, 2) provide *open* web service for energy disaggregation to ordinary consumers. Our service does not require any expertise on data mining or machine learning, and customers can easily obtain energy disaggregation results using either their own historical load data from smart meters deployed by the utility or their own plug-and-play power meters to transmit load data to our web service in real-time.

II. A Plug-and-Play Disaggregation Approach

In this section, we establish a simple and universal optimization model to 1) recover the on/off states of each appliance

978-1-4799-4274-9/14 $31.00 © 2014 IEEE
DOI 10.1109/ICDMW.2014.19

IEEE computer society

along the timeline and 2) estimate its energy consumption during the considered time interval, based on a simple power model of appliances.

Without loss of generality, we assume that a house is equipped with N appliances. Then, only referring to readily-available information of appliances, a simple *power model* can be constructed by three power related vectors:

- a *stand-by power vector* that represents appliances' stand-by powers[1] as
$$I := [I_1, I_2, \cdots, I_N]^T \tag{1}$$
- a *rated power vector* that represents appliances' rated powers[2] as
$$P := [P_1, P_2, \cdots, P_N]^T \tag{2}$$
- a *power deviation vector* that represents appliances' power deviations[3] as
$$\Theta := [\Theta_1, \Theta_2, \cdots, \Theta_N]^T \tag{3}$$

Then, an *aggregated power vector* can be constructed to represent the aggregated power readings generated under the above power model (from time $t = 1$ to K) as

$$X := [X_1, X_2, \cdots, X_K]^T \tag{4}$$

Furthermore, we treat the on/off states of the N appliances (from time $t = 1$ to K) as variables and denote them with a *state matrix* as

$$S := \begin{bmatrix} S_1^{(1)} & S_2^{(1)} & \cdots & S_K^{(1)} \\ S_1^{(2)} & S_2^{(2)} & \cdots & S_K^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ S_1^{(N)} & S_2^{(N)} & \cdots & S_K^{(N)} \end{bmatrix} \tag{5}$$

As a universal property of appliances' activities, the *sparsity* of appliances' state switching events along the timeline has been recognized [3]. By associating this important property with the pre-defined power model, we establish the following sparse switching event recovery (SSER) optimization model to recover the state matrix from the aggregated power vector.

$$\begin{aligned} \min_{S} \quad & \mathbf{TV}(SD) \\ \textit{s.t.} \quad & X - S^T(P+\Theta) - (\mathbf{1}-S)^T I \le \mathbf{0}, \\ & S^T(P-\Theta) + (\mathbf{1}-S)^T I - X \le \mathbf{0} \end{aligned} \tag{6}$$

in which $\mathbf{TV}(A)$ denotes the *total variation* of matrix A, calculated by

$$\mathbf{TV}(A) := \sum_i \sum_j |A_{i,j}| \tag{7}$$

and D is a N-by-$(N-1)$ *difference matrix* defined by

$$D := \underbrace{\begin{bmatrix} -1 & & & & \\ 1 & -1 & & & \\ & 1 & \ddots & & \\ & & \ddots & -1 & \\ & & & 1 & -1 \\ & & & & 1 \end{bmatrix}}_{K-1} \tag{8}$$

and $\mathbf{0}$ is a K dimensional all-zero vector, $\mathbf{1}$ is the N-by-K all-one matrix.

In the above SSER model, by using TV minimization, we actually minimize the total number of state changes (switching events) along the timeline. As to the constraints, they capture a simple fact that the values of aggregated power readings should fall into the inner products of all appliances' states and their corresponding upper and lower power bounds. By solving (6), we can recover the state matrix S, *i.e.*, the on/off states of each appliance along the timeline. Then, the energy consumption of each appliance can be estimated with its rated power. Meanwhile, based on the information of power deviation, we can also provide the upper and lower energy consumption bounds of individual appliance. Eventually, we achieve the initial objective of energy disaggregation.

The detail about the theoretical analysis and validated efficiency of SSER model in solving energy disaggregation can be found in our technical report [10]. Note that, in the SSER model, the stand-by power and rated power of an appliance can be learnt from users' manual, technical specification or public web sites [1], [8], and the power deviation can be easily measured or estimated (and we have showed that SSER mode is robust to inaccurate estimation of power deviation in [10]). Therefore, no particular measuring or training work is needed, making it a plug-and-play approach.

III. Architecture and Functions

A. System Architecture

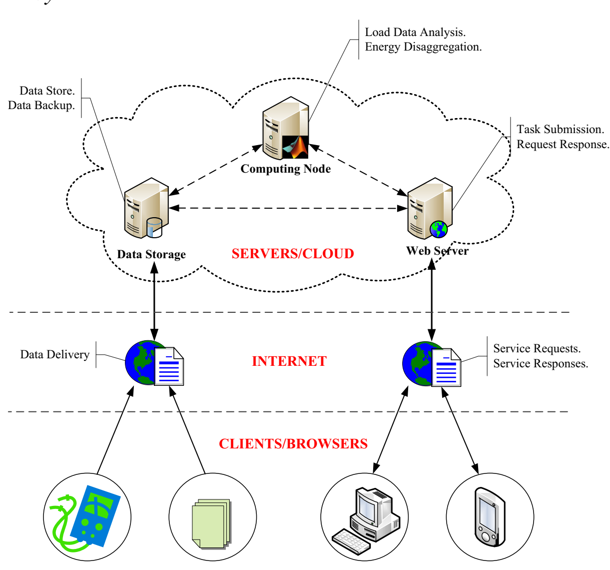

Fig. 2. System architecture of Smart Saver

[1]The stand-by power of an appliance refers to its power usage while it is switched off but unplugged.

[2]The rated power refers to the mean value of real power consumption of an appliance, and is usually included in the appliance's specification.

[3]The power deviation refers to the (approximate) maximum difference between the actual power and rated power, which can be measured/estimated from the power readings.

The logical architecture of Smart Saver is illustrated in Fig. 2. As shown in the figure, Smart Saver is a hybrid system combining the Client/Server (C/S) and Browser/Server (B/S) models. The core of the system is composed of a Data Storage, a Computing Node and a Web Server, which play roles of data storage and backup, load data analysis and energy disaggregation, and task submission and request response, respectively. The three components can be separated or combined, and can be virtualized on the *Cloud* as well (as most commercial energy disaggregation providers have done). On the consumer side, they can either upload real-time load data from their own plug-and-play meters via specific gateways, or submit historical power readings obtained from power suppliers in a point-and-click way via the browser. The load data, service requests and responses are transmitted over the Internet.

B. Functional Design

Under the above system architecture, Smart Saver includes three basic functions.

1) Load Data Collection: The users can connect their own power meters embedded with communication components to our database and upload their real-time load data to the Data Storage over the Internet. For such users, they can not only be served with energy disaggregation but also monitor the real-time states of their appliances. Alternatively, the users can also submit historical load data (i.e., the time series of meter readings) obtained from power suppliers to the Data Storage via the Web Server.

2) Load Data Analysis and Mining: With customers' requests of energy disaggregation or appliance state monitoring submitted via the Web Server, the Computing Node can run specified energy disaggregation algorithms over the aggregated load data to mine the individual appliance's energy consumption or states. The disaggregated energy consumption or recovered appliances' states are returned to the Web Server.

3) Load Data Visualization: Via the browser on the PC or mobile terminals, customers can either read their total energy consumption of multiple appliances directly from the Data Storage, or perform energy disaggregation for individual appliance by submitting disaggregation requests to the Computing Node. Both services use graphical representations to return user-friendly results.

Overall, Smart Saver can provide both *online* and *offline* energy disaggregation services. In the former, the real-time load data from customers are transmitted to the Data Storage, and results of energy disaggregation as well as appliances' states are returned. In the latter, the historical load data from customers are uploaded to the Data Storage via the Web Server, and results of energy disaggregation within the considered time duration are returned. These two kinds of services are expected to fulfill the demands of energy disaggregation under different scenarios for most customers.

IV. Implementation and User Cases

A. Online Load Data Collection

As mentioned before, the customers should be able to connect their power meters to our Data Storage and upload the load data in real-time. In our implementation, we choose the off-the-shelf products developed by Current Cost [4]. As shown in Fig. 3, the *IAMs* measure the power consumption of (individual or multiple) running appliances; the *EnviR* receives the data from IAMs, displays the power readings on the screen, and retransmits the load data to the gateway (the laptop in the figure); the laptop runs our client application (named as *CC Data Collector*) and forwards load messages to our remote Data Storage in real-time. On the Data Storage side, the received load messages (in XML format) are parsed and inserted into our predefined database.

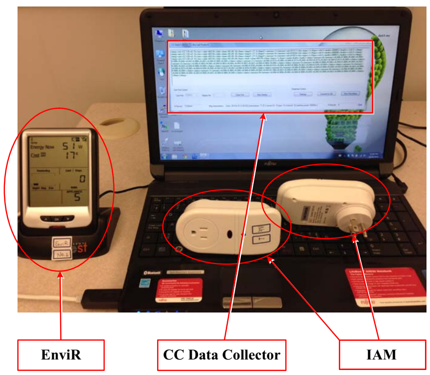

Fig. 3. Plug-and-play power meters, gateway and client application used for online load data collection

In such a way, the real-time load data of consumers can be transmitted to the Data Storage, based on which we can support online energy disaggregation and appliance state monitoring. For the users who have no power meters or do not want to connect their meters to our Data Storage, they can upload their historical data to use our offline service, which will be introduced later.

B. Energy Disaggregation Algorithm

A parallel local optimization algorithm (PLOA) was proposed to solve the SSER model introduced in Section II, and the theoretical analysis and practical validation of the algorithm can be found in [10]. We implement PLOA on our Computing Node that responds to the disaggregation requests submitted via the Web Server. With appliances' power information (provided by the user along with the energy disaggregation requests) and their aggregated power readings, the PLOA can recover the states of individual appliances along the timeline and predict the energy consumption of individual appliances within the specified time period. The resulted states matrix and disaggregated values are returned to the customers via the Web Server.

Note that, besides PLOA, other tasks, such as data cleansing, re-sampling, and smoothing, are performed on the Computing Node. Furthermore, to make the system extendable, we provide an interface so that any other algorithms using only appliances' aggregated data and similar power model can be added into the library on the Computing Node.

C. User Interface

A website is established on the Web Server, which enables consumers to use Smart Saver and perform energy disaggregation in a point-and-click way. Here we show some screenshots of our website and illustrate the major functions and services provided by Smart Saver.

1) Energy Consumption Display: For the users who have connected their power meters to our Data Storage, they can view the real-time energy consumption from each sensor (like IAM). As illustrated in Fig. 4, the energy consumption of appliances from different sensors is shown.

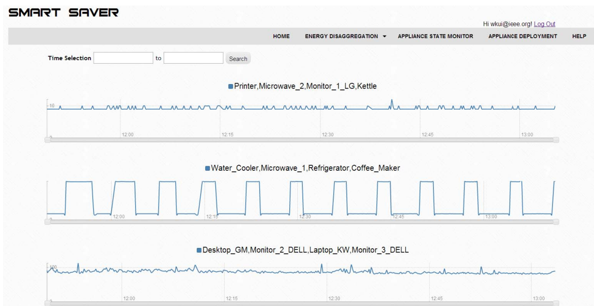

Fig. 4. Website screenshot: aggregated energy consumption display

2) Online and Offline Energy Disaggregation Request: For the users who have connected their power meters to our Data Storage, as shown in Fig. 5-(a), they can submit requests for online energy disaggregation, by providing the sensor id (channel id and collector id) and time interval. For the users using offline energy disaggregation, as shown in Fig. 5-(b), they can submit requests for offline analysis, by completing appliances' information and uploading a historical data file containing records of aggregated energy consumption.

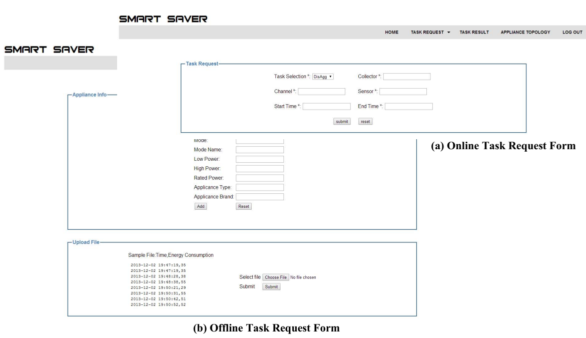

Fig. 5. Website screenshot: request forms for online and offline analysis

3) Energy Disaggregation Results: Once the online/offline energy disaggregation is completed by the Computing Node, the results are returned to the Web Server. The energy consumption of individual appliances and their percentage in the total consumption are shown on the web page of customers, as illustrated in Fig. 6. Furthermore, for online energy disaggregation, the real-time on/off states of individual appliances will also be shown on the web page.

V. Conclusions and Future Work

In this demo we show Smart Saver, a consumer-oriented, user-friendly web service for energy disaggregation. By establishing sparse switching event recovery (SSER) model, we provide energy disaggregation service by only considering the power information of appliances. Using a Data Storage, a Computing Node and a Web Server as the core, we design the architecture and major functions of the system. Based on the logical design, we implement Smart Saver which offers both online and offline energy disaggregation services to consumers. Some user cases of our website are shown in the demo.

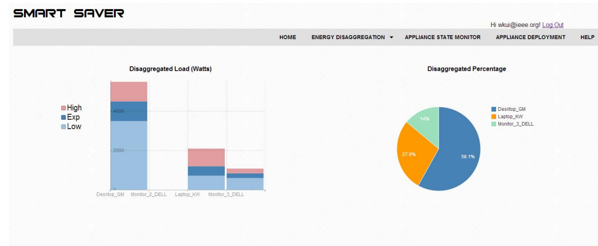

Fig. 6. Website screenshot: energy disaggregation results

VI. Demonstration Plan

The operation and performance of Smart Saver have been captured in a video available at http://youtu.be/eVhg5DwEcDI or http://v.youku.com/v_show/id_XNzMyNzI5MTE2.html.

For the demonstration, we will show how to use Smart Saver with the web browser. Firstly, how appliances are initially deployed with sensor devices as well as how to add, update and delete appliances from the Data Storage will be demonstrated. Then, under a pre-defined appliances' deployment, we will show how to submit online and offline requests to achieve energy disaggregation.

References

[1] EPA. Home product finder. http://www.energystar.gov/productfinder/, 2014. [Online; accessed in 18-June-2014].

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

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

[4] C. C. Inc. Current cost products. http://www.currentcost.com/, 2014. [Online; accessed 20-June-2014].

[5] H. Kim, M. Marwah, M. F. Arlitt, G. Lyon, and J. Han. Unsupervised disaggregation of low frequency power measurements. In *SDM*, pages 747–758, 2011.

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

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

[8] L. B. N. Laboratory. Standby power summary table. http://standby.lbl.gov/summary-table.html, 2014. [Online; accessed in 10-June-2014].

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

[10] G. Tang, K. Wu, J. Lei, and J. Tang. Plug and play! a simple, universal model for energy disaggregation. http://arxiv.org/pdf/1404.1884v1.pdf, 2014.

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