INDiC: Improved Non intrusive load monitoring using load Division and Calibration

Author1
Indraprastha Institute of Information Technology
India

Author2
Twentieth Century Fox
Springfield, USA
Email: homer@thesimpsons.com

James Kirk
and Montgomery Scott
Starfleet Academy
San Francisco, California 96678-2391
Telephone: (800) 555–1212
Fax: (888) 555–1212

***Abstract*—Non-Intrusive appliance load monitoring (NIALM) is the process of disaggregating the overall electricity usage into constituent appliances. In this paper we extend the Combinatorial Optimization (CO) approach for disaggregation, which was originally proposed in the seminal work on NIALM, in following two ways: 1) Breaking the problem into subproblems and reducing the state space; 2) Applying additional constraints backed by sound domain expertise. We evaluate our approach using REDD dataset and show practical problems which need to be solved while dealing with the dataset. We also propose a metric for evaluating NILM, which we believe overcomes many shortcomings of commonly used metrics.**

I. Introduction

- Motivate the importance of energy consumption in building
- Motivate that appliance level information is crucial - detailed feedback and optimized decision making [1]
- Challenges with getting appliance level information - introduce NIALM [2]
- introduce your proposed approach
- Enumerate the contributions

Primary contributions of our work are:

- Fill it up with 2-3 crisp points

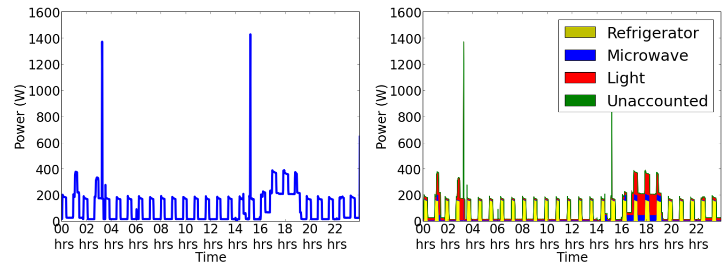
Open source implementation of the proposed work is released for comparative analysis with other NIALM approaches as an IPython notebook[1]. We believe this is the first extensive release of a generic NIALM

II. Related work

NIALM has been well studied in the recent past and survey papers [3], [4], [5] present its classification across various dimensions. Following are three important classification dimensions:

- **Frequency of data collection**: Approaches such as harmonic analysis require data to be sampled at more than a thousand samples a second. Whereas approaches
- **Supervised/Unsupervised**:

[1] http://www.ipython.org



(a) Aggregate Power (b) Disaggregated Power

Fig. 1: The process of disaggregation

When you do the comparison, bring up how is your work different rather than just saying X did A and Y did B.

- Classification of different NIALM approaches - High/Low frequency, Time/Frequency domain analysis, supervised/unsupervised [3], [4], [5].
  For a more detailed overview the reader is referred to the above mentioned survey papers.
- Discuss the modeling approaches that are used
  - Additive Factorial HMM
  - Difference HMM [6]
- Datasets used: Recent datasets have spurred this field
  - REDD [7]
  - Blued [8]
  - Smart* [9]

III. NIALM

NIALM is the process of disaggregating the total electrical load into constituent appliances [2]. A typical NIALM setup involves instrumenting the power mains with smart meters and the aim is to attribute the whole home electricity load perceived at the smart meter into different appliances. Figure 1 shows the house mains disaggregated into 3 appliances.

We use some terminologies from previous work and introduce some which are needed for our analysis in Table I. Based on these terminologies the NIALM problem in supervised setting may be formulated as predicting the power sequence for n^{th} appliance, given by y^n, given the labeled power sequence

TABLE I: Terminologies

Symbol	Meaning
$t \in 1,..T$	Time slice
$n \in 1,..N$	Appliance number
$\theta^n = \{\theta_1^n, ..., \theta_T^n\}$	Labeled power sequence for each appliance
$\theta^{M_1} = \{\theta_1^{M_1}, ..., \theta_T^{M_1}\}$	Labeled power sequence for Mains 1
$\theta^{M_2} = \{\theta_1^{M_2}, ..., \theta_T^{M_2}\}$	Labeled power sequence for Mains 2
$x = \{x_1, .., x_T\} = \theta^{M_1} + \theta^{M_2}$	Labeled aggregate power sequence
$e^{M_1} = \{e_1^{M_1}, ..., e_T^{M_1}\}$	Noise power sequence for Mains 1
$e^{M_2} = \{e_1^{M_2}, ..., e_T^{M_2}\}$	Noise power sequence for Mains 2
$e = \{e_1, ..., e_T\}$	Aggregate noise power sequence
$y^n = \{y_1^n, ..y_T^n\}$	Predicted power sequence for n^{th} appliance
$k \in 1,..K$	Appliance power state
$z^n = \{z_1^n, ..z_T^n\}$	Appliance state sequence,$z_i^n \in [1, ..K]$
$z_{i,k}^n \in 0, 1$	Whether n^{th} appliance is in k^{th} state at time i
$\mu^n = \{\mu_1^n, ..\mu_K^n\}$	Power draw by n^{th} appliance in k^{th} state
$Mapping[n] \in 1, 2$	Mapping of n^{th} appliance to Mains
B	List of background appliances (which can run without human intervention) eg. refrigerator
F	List of foreground appliances (which are operated by humans) eg. light, microwave
$Downsample(s, filter, interval)$	Function to downsample a timeseries s to an $interval$ according to specified $filter$
$Align([s^1, ..s^n], method)$	Function to align n timeseries accounting for missing data using specified $method$
$Sort([s^1, ..s^n], rule, order)$	Function to sort n timeseries according to $rule$ in specified $order$
$Contiguous\_Below\_Mean(s, min\_period)$	Function to find contiguous period where time series s is below its mean for atleast min_period
$Step\_Changes(s, threshold)$	Function returning magnitude and times of step changes occurring in time series s, whose absolute value is greater than $threshold$

for each appliance θ^n and the total aggregate power sequence x.

Combinatorial optimization: If n_{th} appliance is in k_{th} state at , the total power consumption at a given time slice is given by: $\sum \lim_{n=1}^{N} \mu^n$

NIALM using combinatorial optimization (CO) resembles subset sum problem [10]. The aim is to minimize the difference of total observed power from the sum of various possible subsets coming from various combinations of appliances in different states. For each appliance we assume **K** states and at a given time, an appliance can only be in a single state. This is given as: $z_{t,k}^n \in \{0, 1\}$ and,

$$\sum_{k=1}^{k=K} z_{t,k}^n = 1$$

. The power drawn by n^{th} appliance in k^{th} state is given by:

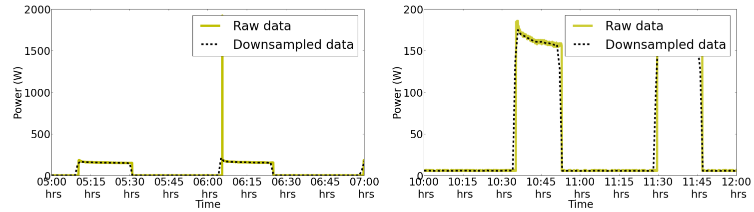
$$\mu^n = \{\mu_1^n, ..\mu_K^n\}$$

Thus, CO can be formulated as:

$$z_t = argmin_{z_t} |x_t - \sum_{n=1}^{N} \sum_{k=1}^{K} z_{t,k}^n \mu_k^n|$$

Correspondingly the power draw by n^{th} appliance is given by: $y^n = \{\mu_{z_1^n}^n, ..\mu_{z_T^n}^n\}$

This problem is NP complete and the state space is K^N, which is exponential in the number of appliances used. Owing to the exponential nature of the state space and the fact that the algorithm requires all appliances be known, this approach has not been thoroughly studied in the past. We chose to use this as a proof of concept of our contributions.



(a) Filtering startup transients (b) Filtering voltage fluctuations and oscillations

Fig. 2: Effect of downsampling appliance data

- Statespace is
- We assign different loads to different mains, N_i loads to $Mains_i$, $\sum_{1}^{p} N_i = N$. Now different state spaces are K^{N_1}.... We can define the overall state space as $\max K^{N_i}$
  As a practical example, two mains, 20 appliance, state space before = 2^{20}. After = 2^{10}. Exponential reduction in state space.

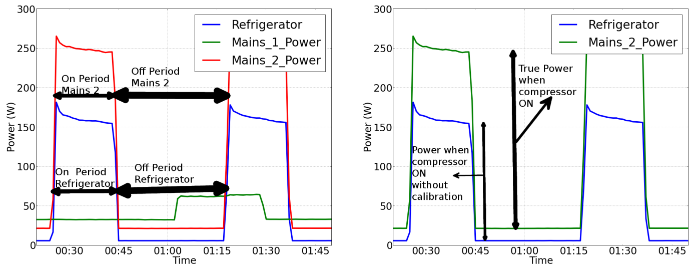
Highlight what is the simplification you are bringing forth.

IV. INDIC NIALM

In this section we explain the various steps involved in DaCo-NIALM which is shown in Figure **??** or Algorithm 1.

Downsample and align raw data: While performing Combinatorial Optimization it is desired that transients and fluctuations in the power signal are filtered. The transients occur due to the high starting current of the appliance, whereas the fluctuations are a consequence of minor voltage fluctuations and oscillatory nature of loads. Figure 2a and Figure 2b show how starting current and voltage fluctuations can be filtered by downsampling. Further realignment amongst the appliance level data and mains level data is needed owing to different frequency of data collection and missing data.

Assigning Loads to Mains: This step aims to identify the mapping between appliances and mains. Based on domain expertise we label the appliances in a home into background (loads which run independently throughout the day without user interference) such as refrigerator, and foreground (loads which are highly correlated with human usage) such as stove. Background loads are easier to detect since they are On even during periods of low human activity. Also loads with higher mean power consumption are easier to identify and thus we sort background loads based on power in descending order. For all such background loads we see if the mean power of the appliance is greater than mean power of any mains for all time instances. If so, we can safely assign the appliance to the other mains. If this step is unable to provide conclusive evidence we look at the periodicity associated with such background loads during periods of low or no human activity (such as night time). Figure 3a shows how based on refrigerator duty cycle it is mapped to Mains 2. On similar lines assignment of foreground loads can be done.



(a) Assigning refrigerator to Mains 2 since refrigerator power > Mains 1 power and on basis of duty cycle

(b) Calibrating Refrigerator Power consumption

Fig. 3: Calibrating and assigning refrigerator to Mains 2

Appliance Power Calibration: Power measured by appliance level meters may need calibration due to the following reasons:

- Difference in measurement of different measurement instrument (diagram showing CC, ZWave, etc)
- Fluctuation in voltage cause power to fluctuate as well
- Missing meta data, whether real or apparent power is being measured at appliance level

Since different hardware is used for measuring appliance and mains data there may be a need to calibrate the two. Since mains data is usually collected using better precision hardware, we keep mains data as a reference and calibrate appliance data against it. In practice we found appliance level monitors to usually provide only real power whereas the mains monitors can provide much more like reactive and active power. Like the previous step, time instances when an appliance in a particular mains is single used are identified. The ratio of mains and appliance power step changes occurring this window serve as the calibration factor for that appliance. Further each appliance power is corrected with the corresponding calibration factor.

Clustering to identify appliance states:

1) Step changes occurring in Mains vs Appliances
2) Isolating single appliance usage We use [11] to run our clustering

Appliance states identification using clustering (Possibly talk about unbalanced data, but leave it for future work) State space creation Applying CO for different mains Find energy distribution by appliance and assign weights (To be used in results)

A. Load assignment

Draws inspiration from work by Parson et. al [6]. From prior knowledge we divide the loads into two different categories: Periodic such as refrigerator and non periodic such as Television.

V. Evaluation

A. About Dataset

We use Reference Energy Disaggregation Data Set (REDD) [7] for validating our algorithms. This dataset contains power

Algorithm 1: INDiC

Input: $x, \theta^n, \theta^{M_1}, \theta^{M_2}, B, F$
Output: y^n, μ_k^n

Align and Downsample
for $n \in 1, N$ **do**
  $\theta^n \leftarrow Downsample(\theta^n, mean, 1\, minute)$
$\theta^1, ..\theta^n, \theta^{M_1}, \theta^{M_2} \leftarrow Align([\theta^1, ..\theta^n, \theta^{M_1}, \theta^{M_2}], forward\, fill)$
Appliance to Mains mapping
$s \leftarrow Sort([\theta^1, ..\theta^n], peak\, power)$
$l^1 \leftarrow Contiguous\_Below\_Mean(\theta^{M_1}, 2\, hours)$
$l^2 \leftarrow Contiguous\_Below\_Mean(\theta^{M_2}, 2\, hours)$
for $Appliance\, n \in s$ **do**
  if $\theta^n > \theta^{M_1}$ $for\, any\, t \in 1, T$ **then**
    $Mapping[n] = 2$
  else if $\theta^n > \theta^{M_2}$ $for\, any\, t \in 1, T$ **then**
    $Mapping[n] = 1$
Find consecutive time periods ($low\_activity$) of low or minimal electrical activity
Sort background loads in decreasing order of peak power
for *load in sorted list of background loads* **do**
  if *load > either Mains during low_activity* **then**
    Assign *load* to other Mains
  else
    Find duty cycle of *load* and compare with both Mains during *low_activity*
    Assign *load* to the Mains having power step up and down at same time as that of *load*
  Subtract *load* from assigned Mains
Sort foreground loads in decreasing order of peak power
for *load in sorted list of foreground loads* **do**
  if *load > either Mains during entire timeseries* **then**
    Assign *load* to other Mains
  else
    Find time windows when *load* > 100 W and find step changes occurring and their *times* in *load* during these time windows
    Assign *load* to the Mains where similar step changes are found in same *times* during the time windows
  Subtract *load* from assigned Mains
for *load in sorted list of loads* **do**
  Normalize power consumption of each load in case of voltage fluctuation
  Apply additive or multiplicative calibration to load based on step change occurring in *load* and corresponding Mains
  Using prior knowledge about appliances apply clustering techniques to learn different states for loads
Create state space based on NILM algorithm
Apply NILM algorithm on each Mains and set of loads in it
return *Disaggregated mains power data per appliance*

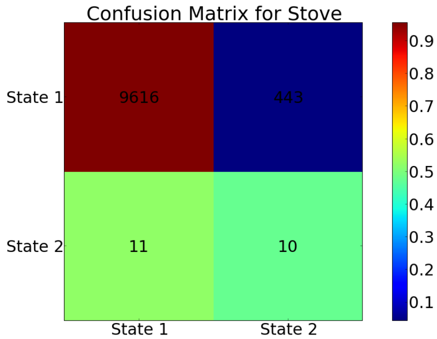


Fig. 4: Confusion Matrix showing predicted state accuracy for Stove

and voltage data for mains (2 phases) as well as appliances from 6 homes in Boston area collected in the summer of 2011. The data is made available as raw, high frequency (sampled at 15 KHz) and low frequency (Mains at 1 Hz, appliances at 0.3 Hz). Considering the practical implications of residential smart meter installation, we believe that low frequency data represents the most realistic scenario and thus we use this data for analysis.

B. Evaluation Metric

Commonly used metrics such as accuracy, sensitivity and specificity can be misleading when applied to NIALM. It can be seen from Figure 4 that since stove is mostly in state 0 (Off), accuracy will be largely decided by accuracy for this state. This does not tell how badly the prediction is misclassifying state 1 (On). Thus, we use the following metrics which have been used in the past work [6], [7]:

- **Mean Normalized Error (MNE %)**: Normalized error in the energy assigned to an appliance n over time period T, given by

$$MNE(n) = \frac{|\sum_{t=1}^{T} \theta_t^n - \sum_{t=1}^{T} y_t^n|}{\sum_{t=1}^{T} \theta_t^n}$$

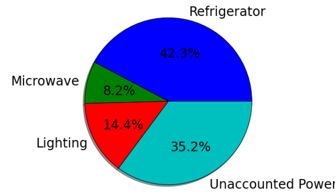
- **RMS Error (RE Watts)**: RMS error in power assignment to an appliance n per time slice given by

$$RE(n) = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(\theta_t^n - y_t^n)^2}$$

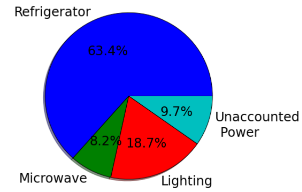
Since both these quantities represent error, the lesser they are the better is the prediction.

C. Empirical Analysis

We performed empirical analysis on REDD dataset Home 2, which consists of 11 channels (including 2 mains and 9 appliances). We believe that the same analysis can be easily repeated across multiple homes. We aligned the data and downsampled it to one minute using mean filter. Since we had two weeks of clean data, we used the first week as the train set and the second week as the test set. We used INDiC algorithm described earlier to assign loads to different mains and calibrate appliance data. Table II shows the assignment of loads to different mains. Further this table also shows the learnt power states of these appliance via k-means++ clustering (which for several reasons is considered better than k-means) [11]. Refrigerator and lighting showed significant difference in power states post calibration. Based on prior experience and appliance circuity, we believe that since only these two appliances needed calibration, it may be a case that the appliance level monitor measured real power instead of apparent power. These loads constitute a major chunk of Mains 2 power. Figure 5 shows the reduction in unassigned power due to calibrating these two appliances. Two loads - washer dryer and disposal did not have significant usage and we chose not to consider them in the analysis.



(a) Without applying calibration



(b) After applying calibration

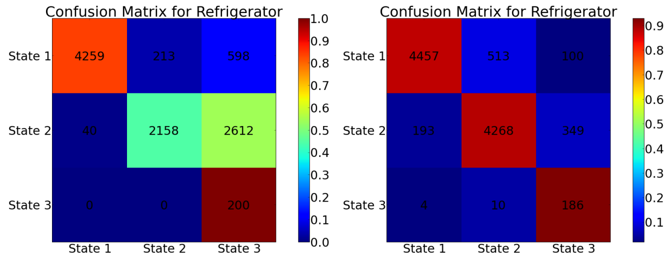
Fig. 5: Mains 2 Break down by load

TABLE II: Calibration Factors, Mains Assignment and States

Appliance	Mains	States Power (W)	States Power (W)
		Pre calibration	Post calibration
Refrigerator	2	7,162,423	9,210,423[2]
Microwave	2	10,832,1730	10,832,1730
Lighting	2	9,96,156	10,110,178
Dishwasher	1	0,256, 1195	0,256, 1195
Stove	1	0,374	0,374
Kitchen	1	5,727	5,727
Kitchen 2	1	1,204,1032	1,204,1032

To show the significance of load division and calibration, we applied Combinatorial optimization on the test set, considering 4 possible cases: i) no calibration, no load division; ii) no calibration, load division; iii) calibration, no load division; iv) calibration, load division. These results are presented in Table III. For the overall dataset, it can be seen that MNE reduces from 187% to 39%, RE reduces from 478 W to 168 W, after applying INDiC. All appliances show reduction in MNE and RE after applying INDiC. However, there is significant improvement in correctly predicting refrigerator and lighting. Figure 6 shows the confusion matrix for refrigerator prediction pre and post applying INDiC. It can be seen that after applying INDiC there is a vast improvement in predicting refrigerator's state 0 and 1.

We had used Combinatorial Optimization which is the simplest NIALM technique to show the improvements which can be made by load division and appliance calibration. We believe that using state of the art NIALM algorithms will improve the results by leaps and bounds.



(a) Without applying DiCaCo (b) After applying DiCaCo

Fig. 6: Confusion Matrix for refrigerator disaggregation

TABLE III: Mean Normalized Error and RMS error with and without DiCaCo NIALM

	Without Recalibration				With Recalibration			
	Without Load Division		With Load Division		Without Load Division		With Load Division	
Appliance	R.E. Watts	M.N.E. %	R.E. Watts	M.N.E. %	R.E. Watts	M.N.E. %	R.E. Watts	M.N.E. %
Refrigerator	136	109	71	32	130	95	59	21
Microwave	102	98	97	110	104	97	96	109
Lighting	51	164	48	195	44	83	38	60
Dishwasher	406	2947	63	100	377	2517	63	100
Stove	77	1191	36	281	75	1118	36	281
Kitchen	64	182	58	168	69	196	58	168
Kitchen 2	95	267	91	117	92	230	91	117
Overall	478	187	161	58	450	157	168	39

VI. Conclusion

The conclusion goes here. We also provide mains load assignment of all 6 homes from REDD to further the research in this direction.

VII. Future Work

- Applying model on noisy datasets
- 2 D CO (when Real and Reactive Power are known)
- Factoring in Time of Day etc.
- Factoring in Appliance Correlation
- Factor in switch continuity, essentially leads to Factorial HMM
- Distributed NILM
- Adaptive Learning

Acknowledgment

The authors would like to thank TCS Research and Development for supporting the first author through PhD. fellowship. We would also like to thank NSF- DEITy for funding the project.

References

[1] S. Darby, "The effectiveness of feedback on energy consumption," *A Review for DEFRA of the Literature on Metering, Billing and direct Displays*, vol. 486, 2006.

[2] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

[3] K. Carrie Armel, A. Gupta, G. Shrimali, and A. Albert, "Is disaggregation the holy grail of energy efficiency? the case of electricity," *Energy Policy*, 2012.

[4] A. Zoha, A. Gluhak, M. A. Imran, and S. Rajasegarar, "Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey," *Sensors*, vol. 12, no. 12, pp. 16 838–16 866, 2012.

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

[6] O. Parson, S. Ghosh, M. Weal, and A. Rogers, "Non-intrusive load monitoring using prior models of general appliance types," in *26th AAAI Conference on Artificial Intelligence*, 2012.

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

[8] A. Filip, "Blued: A fully labeled public dataset for event-based non-intrusive load monitoring research."

[9] S. Barker, A. Mishra, D. Irwin, E. Cecchet, P. Shenoy, and J. Albrecht, "Smart*: An open data set and tools for enabling research in sustainable homes," in *The 1st KDD Workshop on Data Mining Applications in Sustainability (SustKDD)*, 2011.

[10] S. Martello and P. Toth, *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Inc., 1990.

[11] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.