Compressive Non-Intrusive Load Monitoring

Shikha Singh
Indraprastha Institute of Information Technology, Delhi, India
shikhas@iiitd.ac.in

Angshul Majumdar
Indraprastha Institute of Information Technology, Delhi, India
angshul@iiitd.ac.in

Stephen Makonin
Simon Fraser University
Vancouver, BC, Canada
smakonin@sfu.ca

ABSTRACT

In non-intrusive load monitoring (NILM), an increase in sampling frequency translates to capturing unique signal features during transient states, which, in turn, can improve disaggregation accuracy. Smart meters are capable of sampling at a high frequency (typically 20kHz). However, transmitting signals continuously would choke the network bandwidth. Given the deployment of millions of smart meters which communicate over a wireless wide-area network (WAN), utilities can only collect power signals at very low frequencies. We propose a compressive sampling (CS) approach. After measuring the high-frequency power signal from a smart meter will be encoded (by a random matrix) to very few samples making the signal suitable for WAN transmission without choking network bandwidth. CS guarantees the recovery of the high-frequency signal from the few transmitted samples under certain conditions. This work shows how to simultaneously recover the signal and disaggregate it; hence, the name Compressive NILM.

CCS CONCEPTS

• **Information systems → Information retrieval**.

KEYWORDS

non-intrusive load monitoring, NILM, energy disaggregation, compressed sensing, compressive sampling, sparse coding

ACM Reference Format:
Shikha Singh, Angshul Majumdar, and Stephen Makonin. 2020. Compressive Non-Intrusive Load Monitoring. In *The 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '20), November 18–20, 2020, Virtual Event, Japan.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3408308.3427613

1 INTRODUCTION

Energy disaggregation is the task of estimating the energy consumption of individual electrical appliances given the total consumption recorded by the smart-meter. It is a single channel (smart-meter) blind source (appliances) separation problem. This makes the problem highly underdetermined in nature – one equation (smart-meter consumption) and many variable (appliance consumption). Therefore the problem has infinitely many solutions.

When the power signal is of sufficiently high frequency, integer programming based approaches provide a feasible solution [2, 20]. Similarly factorial hidden Markov model (FHMM) is used to disaggregate appliance loads from high frequency samples [12, 16]. The performance of such techniques degrades when the sampling frequency is reduced. Sparse coding approaches yield somewhat better results at low-frequencies [9, 11]; however, even with sparse coding, higher frequencies translate to better results.

Smart-meters can sample at high frequencies, but higher frequencies mean generation of more data. Transmitting this data from the building smart-meter to the cloud at the utilities consumes some bandwidth; higher the sampling frequency higher is the bandwidth consumed. Note that, it is not only one building that would be transmitting this data, all the buildings would be transmitting it; in such a scenario it is likely the entire network bandwidth will be consumed for only transmitting power signals! To keep the network usage at check, the smart-meter transmits the signal at low-frequencies (even though it is capable of sampling at high frequencies).

Typically it is expected that energy disaggregation would be offered as a service by the utilities. However, since the utilities will have access to only low-frequency information, the disaggregation accuracy is likely to suffer. To bridge the gap between high-frequency sampling and low-frequency transmission we propose a compressed sensing / compressive sampling (CS) approach [4, 5, 7]. We project the high frequency signal to a lower dimension embedding by a random projection matrix. The lower dimensional signal will emulate a low frequency signal which can be then transmitted. The random projection can be easily integrated into hardware [6, 21]. Under certain conditions, such a lower dimensional embedding approximately preserves the information of the high frequency signal and can be recovered using sparsity promoting techniques like $\ell 1$-minimization [10] or matching pursuits like algorithms [18].

This work extends the traditional compressed sensing dictates (e.i., recovering the signal) by adding simultaneous disaggregation as part of the recovery process. Our formulation is based on the dictionary learning approach [19] (the same technique used in sparse coding [9, 11]).

The paper will be organized into several sections. We will discuss the basics of CS in the following section. In Section 3, we describe our proposed formulation. The results will be detailed in Section 4. Finally, the conclusions of this work will be discussed in Section 5.

2 COMPRESSIVE SAMPLING

Compressed Sensing (CS) studies the problem of solving an underdetermined linear system of equations where the solution is known to be sparse. In practical scenarios, the system is corrupted by noise as well.

BuildSys '20, November 18–20, 2020, Virtual Event, Japan

ACM ISBN 978-1-4503-8061-4/20/11…$15.00
https://doi.org/10.1145/3408308.3427613

$$y_{m\times 1} = A_{m\times n}x_{n\times 1} + \varepsilon_{m\times 1}, m < n \tag{1}$$

The solution x, is assumed to be k sparse ($k \ll m < n$).

For an underdetermined system, there can be infinitely many solutions. Research shows that when the solution is sparse, it is necessarily unique [8]; i.e., there cannot be more than one sparse solution. Further research established that when the number of equations satisfies the following criterion (2), $\ell 1$-minimization can recover the sparse solution.

$$m = ck\log(\tfrac{n}{k}) \tag{2}$$

The $\ell 1$-norm minimization is robust to noise [3]. The recovery is formulated as:

$$\min_{x} \|y - Ax\|_2^2 + \lambda\|x\|_1 \tag{3}$$

CS recovery is not possible for any system of equations A; it is only guaranteed when the so called restricted isometric property (RIP) holds. This condition is expressed as follows:

$$(1-\delta)\|x\|_2^2 \le \|Ax\|_2^2 \le (1+\delta)\|x\|_2^2 \tag{4}$$

Here δ is a small constant. RIP guarantees that the system A behaves as a near isometry. The value of δ dictates how much the system deviates from ideal Isometry. This property is usually satisfied by random matrices for example, restricted Fourier ensembles and matrices drawn from distributions such as Gaussian, Bernoulli, and Binomial.

Practical systems/signals are hardly ever sparse. However, most of them have a sparse representation in some transform domain. For example, images are sparse in discrete cosine transform (DCT) or wavelet, speech is sparse in short time Fourier transform, etc. This phenomenon allows expression of the signal x in terms of transform domain sparse coefficients α,

$$Analysis : \alpha = \Psi x \tag{5a}$$

$$Synthesis : x = \Psi^T\alpha \tag{5b}$$

Here Ψ is the sparsifying transform and the relationships (5) hold for orthogonal ($\Psi^T\Psi = I = \Psi\Psi^T$) and tight-frame ($\Psi^T\Psi = I \neq \Psi\Psi^T$) systems.

For signals that have a sparse representation in the transform domain, the recovery is expressed as follows:

$$\min_{\alpha} \|y - A\Psi\alpha\|_2^2 + \lambda\|\alpha\|_1 \tag{6}$$

Once the sparse coefficients are recovered, the signal is obtained by applying the synthesis equation (5b).

Following (2), note the number of equations/samples needed to recover a signal is directly dependent on the sparse representation α and thereby on the choice of the transform Ψ. For example, if x is an image, the number of corresponding non-zero DCT coefficients will be higher than the corresponding wavelet coefficients making the choice of coefficient a crucial step in CS recovery.

Fixed transforms like Fourier, DCT wavelet have nice mathematical properties but are not known to produce the sparsest representation. In signal processing, it is well known that an adaptive basis (learnt from the signal) produces the sparsest representations. This paved the way for dictionary learning based solutions; starting with the work on K-SVD [1].

In dictionary learning, the sparsity basis is learnt from the data. For example, if the problem involves an image, the sparsity basis is learnt from the patches of the image. The recovery is posed as:

$$\min_{x,D,z_i's} \|y - Ax\|_2^2 + \mu\Big(\underbrace{\sum_i \|P_i x - Dz_i\| + \lambda\|z_i\|_1}_{\text{Dictionary Lxearning}}\Big) \tag{7}$$

Here P_i representation patch extraction operator; D is the basis that is being adaptively learnt from the patches and z_i are the corresponding sparse representations of the patch $P_i x$. In dictionary learning, D replaces the role of Ψ in CS.

When the basis is learnt adaptively the recovery results are far better than of classical CS where the sparsifying basis is fixed. There are many other branches of CS and dictionary learning, but these are not pertinent to us. The interested reader may peruse [14].

3 COMPRESSIVE NILM FORMULATION

We assume that the smart-meter is sampling at the rate of n samples per unit of time (say an hour); but is only allowed to transmit $m < n$ samples in that period. Let $x_{n\times 1}$ represent the signal sampled by the smart-meter. Currently a sub-sampled version of x is transmitted; we propose to embed the high dimensional signal into a lower dimensional representation $y_{m\times 1}$ by a random projection matrix $A_{m\times m}$ (satisfying RIP). This is represented by,

$$y = Ax + \varepsilon \tag{8}$$

It is unlikely that the system will be corrupted by noise, but for the sake of generality, we assume Gaussian noise ε. The problem is to disaggregate the appliance level consumption given the lower dimensional representation y. To do so, a standard NILM training and testing approach is followed.

3.1 Training

In the training phase the individual appliances are metered. For each appliance j, the samples for the i-th unit of time is represented by x_i^j. The complete training data for the i-th appliances is represented by,

$$X^j = \left[x_1^j | x_2^j | \ldots | x_N^j\right] \tag{9}$$

Here we assume N units of time in the training phase.

In our proposition, the utilities do not have access to X^j, but has received a lower dimensional projection of it, given by:

$$Y^j = AX^j + E^j \tag{10}$$

where $Y^j = \left[y_1^j | y_2^j | \ldots | y_N^j\right]$ and $E^j = \left[\varepsilon_1^j | \varepsilon_2^j | \ldots | \varepsilon_N^j\right]$.

Following the work of sparse coding [11], each appliance is modeled by a sparse codebook/dictionary D^j. This is expressed as,

$$X^j = D^j Z^j \tag{11}$$

We reiterate that our work assumes that the disaggregation happens at the utilities server/cloud. Incorporating this model (11) to the data received at the utilities we get:

$$Y^j = AD^jZ^j + E^j \tag{12}$$

Following the work on sparse coding, the training phase requires solving for the dictionaries and the sparse codes (not required during testing) for modeling the appliances. This is expressed as,

$$\min_{D^j, Z^j} \|Y^j - AD^jZ^j\|_F^2 + \lambda \|Z^j\|_1 \tag{13}$$

This (13) is easily solved using alternating minimization of the codebook and the sparse codes. During the update for the codebook, the sparse code is assumed to be constant. The update is given by:

$$\min_{D^j} \|Y^j - AD^jZ^j\|_F^2 \implies D^j = A^\dagger Y^j (Z^j)^\dagger$$

where $(\cdot)^\dagger$ denotes the Moore-Penrose pseudoinverse.

The update for the sparse codes assumes that the codebook is fixed. The update is expressed as,

$$\min_{Z^j} \|Y^j - AD^jZ^j\|_F^2 + \lambda \|Z^j\|_1 \tag{14}$$

This is a standard $\ell 1$-minimization problem that can be solved using any iterative thresholding algorithm.

Note that the solution to the codebook and sparse codes automatically reconstructs the original signals acquired by the smart-meter (11).

3.2 Testing

In the testing/operation stage, the task is to disaggregate the total load acquired by the smart-meter. The total load, as recorded by the smart-meter in a unit of time is x_i. Therefore for all M units of time, the data is expressed as,

$$X = \left[x_1 | x_2 | ... | x_M\right] \tag{15}$$

This (15) is an aggregate of the loads consumed by individual appliances.

$$X = \sum_j X^j \tag{16}$$

Incorporating the sparse coding model (11) into (16) leads to:

$$X = \sum_j X^j = \sum_j D^jZ^j \tag{17}$$

As mentioned before, the sparse codes obtained during the training phase are not useful later on, only the codebooks are used in (17).

In the compressive NILM scenario, the utilities do not have access to the fully sampled data X, but has received its lower dimensional embedding $Y : Y = AX + E$. Incorporating (17) into the data acquisition model leads to:

$$Y = A \sum_j D^jZ^j + E \tag{18}$$

During the testing phase, the codebooks are known; the goal is to estimate the sparse codes Z^j. The solution is obtained by minimizing the following,

$$\min_{Z^j{}'s} \left\|Y - A\sum_j D^jZ^j\right\|_F^2 + \sum_j \|Z^j\|_1 \tag{19}$$

As before, this can be solved using any iterative thresholding algorithm.

Once the sparse codes are solved, the power consumption of individual devices can be obtained using (11). Note that our algorithm automatically reconstructs the signal during training and testing phase. If one is interested in applying some other algorithm, they can run it on the reconstructed data at the server/cloud.

4 EXPERIMENTAL RESULTS

Here we report results on the REDD dataset [13] – a moderate size publicly available dataset for electricity disaggregation. The dataset consists of power consumption signals from six different houses, where for each house, the whole electricity consumption, as well as electricity consumptions of about twenty different devices, are recorded. The signals from each house are collected over a period of two weeks with a high-frequency sampling rate of 15kHz. To prepare ground-truth (high frequency) training and testing data, aggregated and sub-metered data are averaged over a time period of 1 minute.

We assume that the utilities acquire the data once every ten minutes. Given this constraint, we further sub-sample the data to once every ten minutes (sub-sampled data), from 60 samples per hour. In the proposed compressive NILM regime, the once per minute data is projected to a lower dimension of 6 samples in hour using a Bernoulli matrix (compressively sampled data); using a Bernoulli projection matrix ensures that our simulations are hardware friendly.

To compare the performance of our proposed approach, we employ FHMM [12] and sparse coding (SC) [11] on the sub-sampled version of the data. These act as the benchmarks. We use our algorithm to disaggregate from the compressive sampled version of the data. As mentioned before, our proposed method reconstructs the high frequency data in the process; on this reconstructed data we apply FHMM. Note that, there is no point in applying sparse coding on the reconstructed data, since our proposed approach effectively does the same.

Appliance-level Precision and Recall are used as metrics for evaluating the performance [15, 17]. While using such a metrics, we tacitly assume that the appliances are binary-state (ON or OFF) while disregard other operational states.

The results depict the expected trend. Both the techniques SC and FHMM perform similarly for a given sampling frequency. The performance is poor when the sampling is done at regular intervals (sub-sampled data); but with compressive sampling, the performance improves considerably. Our proposed method (Compressive NILM) effectively disaggregates (using SC) and reconstructs simultaneously. The FHMM is applied on the thus reconstructed (higher frequency – once very minute) data.

5 CONCLUSIONS

This is the first work (proof-of-concept) on the topic of employing compressed sensing to balance accuracy and bandwidth for NILM tasks. The results show that the approach is promising.

Table 1: Disaggregation Performance Evaluation (Using Precision/Recall)

Appliance	Sub-Sampled SC	Sub-Sampled FHMM	Compressive NILM	Reconstructed FHMM
Microwave	.53/.34	.55/.32	.70/.58	.71/.55
Kitchen Outlet 1	.30/.11	.27/.10	.37/.15	.35/.13
Kitchen Outlet 2	.33/.11	.32/.11	.45/.15	.40/.14
Furnace.	.75/.61	.78/.59	.86/.69	.87/.66
Washer/Dryer	.73/.54	.74/.53	.82/.64	.85/.62

In the future we would like to extend the work in two ways. First, we would like to push the extents of the compressive sampling and attempt to disaggregate and reconstruct from more aggressive sampling (compared to 10:1 used in this work). Second, we would like to attempt some state-of-the-art deep learning models into the CS framework to improve the disaggregation results.

REFERENCES

[1] M. Aharon, M. Elad, and A. Bruckstein. 2006. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing* 54, 11 (2006), 4311–4322.

[2] M. Z. A. Bhotto, S. Makonin, and I. V. Bajić. 2017. Load Disaggregation Based on Aided Linear Integer Programming. *IEEE Transactions on Circuits and Systems II: Express Briefs* 64, 7 (2017), 792–796.

[3] E. J. Candes and Y. Plan. 2011. A Probabilistic and RIPless Theory of Compressed Sensing. *IEEE Transactions on Information Theory* 57, 11 (2011), 7235–7254.

[4] E. J. Candes and T. Tao. 2006. Near-Optimal Signal Recovery From Random Projections: Universal Encoding Strategies? *IEEE Transactions on Information Theory* 52, 12 (2006), 5406–5425.

[5] E. J. Candes and M. B. Wakin. 2008. An Introduction To Compressive Sampling. *IEEE Signal Processing Magazine* 25, 2 (2008), 21–30.

[6] F. Chen, A. P. Chandrakasan, and V. M. Stojanovic. 2012. Design and Analysis of a Hardware-Efficient Compressed Sensing Architecture for Data Compression in Wireless Sensors. *IEEE Journal of Solid-State Circuits* 47, 3 (2012), 744–756.

[7] D. L. Donoho. 2006. Compressed sensing. *IEEE Transactions on Information Theory* 52, 4 (2006), 1289–1306.

[8] David L Donoho. 2006. For most large underdetermined systems of linear equations the minimal $\ell 1$-norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 59, 6 (2006), 797–829.

[9] Ehsan Elhamifar and Shankar Sastry. 2015. Energy Disaggregation via Learning Powerlets and Sparse Coding.. In *AAAI*. 629–635.

[10] Simon Foucart. 2010. A note on guaranteed sparse recovery via $\ell 1$-minimization. *Applied and Computational Harmonic Analysis* 29, 1 (2010), 97–103.

[11] J Zico Kolter, Siddharth Batra, and Andrew Y Ng. 2010. Energy disaggregation via discriminative sparse coding. In *Advances in Neural Information Processing Systems*. 1153–1161.

[12] J Zico Kolter and Tommi Jaakkola. 2012. Approximate inference in additive factorial hmms with application to energy disaggregation. In *Artificial intelligence and statistics*. PMLR, 1472–1482.

[13] J Zico Kolter and Matthew J Johnson. 2011. REDD: A public data set for energy disaggregation research. In *Workshop on data mining applications in sustainability (SIGKDD), San Diego, CA*, Vol. 25. 59–62.

[14] Angshul Majumdar. 2018. *Compressed Sensing for Engineers*. CRC Press.

[15] Stephen Makonin and Fred Popowich. 2015. Nonintrusive load monitoring (NILM) performance evaluation. *Energy Efficiency* 8, 4 (2015), 809–814.

[16] S. Makonin, F. Popowich, I. V. Bajić, B. Gill, and L. Bartram. 2016. Exploiting HMM Sparsity to Perform Online Real-Time Nonintrusive Load Monitoring. *IEEE Transactions on Smart Grid* 7, 6 (2016), 2575–2585.

[17] Lucas Pereira and Nuno Nunes. 2018. Performance evaluation in non-intrusive load monitoring: Datasets, metrics, and tools—A review. *Wiley Interdisciplinary Reviews: data mining and knowledge discovery* 8, 6 (2018), e1265.

[18] S. K. Sahoo and A. Makur. 2015. Signal Recovery from Random Measurements via Extended Orthogonal Matching Pursuit. *IEEE Transactions on Signal Processing* 63, 10 (2015), 2572–2581.

[19] I. Tošić and P. Frossard. 2011. Dictionary Learning. *IEEE Signal Processing Magazine* 28, 2 (2011), 27–38.

[20] F. M. Wittmann, J. C. López, and M. J. Rider. 2018. Nonintrusive Load Monitoring Algorithm Using Mixed-Integer Linear Programming. *IEEE Transactions on Consumer Electronics* 64, 2 (2018), 180–187.

[21] J. Yoo, S. Becker, M. Monge, M. Loh, E. Candès, and A. Emami-Neyestanak. 2012. Design and implementation of a fully integrated compressed-sensing signal acquisition system. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 5325–5328.