Transform Learning Assisted Graph Signal Processing for Low Rate Electrical Load Disaggregation

Kriti Kumar and M Girish Chandra
TCS Research and Innovation, Bangalore, India.
Email: {kriti.kumar, m.gchandra}@tcs.com

***Abstract*—Separating the individual sources with less differentiating features and random time overlaps in a superposition is a challenging problem that arises in many scenarios. Disaggregating low sampled smart meter data is one such important and interesting use case. In this paper, a novel joint optimization formulation using Transform Learning (TL) assisted with Graph Signal Processing (GSP) is presented to reconstruct the individual operational waveforms of the electrical loads. Data-driven transforms are utilized to learn the individual load characteristics. Treating the transform coefficients as load activations, graph signal smoothness is exploited to estimate the coefficients in the test phase using the coefficients learnt during the training phase. The requisite optimization formulation and the derivation of the necessary update steps are presented. The efficacy of the proposal is demonstrated by the load identification and consumption estimation results obtained for residential load disaggregation, considering both real and simulated data along with comparisons against some of the recent works in this domain.**

***Index Terms*—Transform Learning, Dictionary Learning, Graph Signal Processing, Electrical Load Disaggregation**

I. Introduction

Electrical load disaggregation can be viewed as a source separation problem where, given the aggregate power measurements, it is required to detect the individual loads in operation and estimate their power consumption. Disaggregation results can be used to host a lot of applications which are beneficial to both utilities and consumers [1]. It also helps in reducing the carbon footprint by increasing the awareness among the consumers about their energy consumption. Being an important component of energy management in the emerging power systems, it has been a topic of research in the recent past. A lot of algorithms for load disaggregation or Non Intrusive Load Monitoring (NILM) exist in literature [2]–[6].

Most of the NILM techniques work well when the sampling intervals of the aggregate power measurements is low i.e. $\leq$ 10 seconds. However, when the sampling intervals increases to order of minutes, e.g. 15 or 30 minutes, the disaggregation of loads becomes very difficult [7]. This is because at such low sampling intervals, no prominent events and signatures associated with different loads operation are observed. The current smart meters capture aggregate power measurements at 15 or 30 minute intervals. With their rapid installation, there is a need to process this data to extract useful information about the loads, particularly the high power consuming loads. This information can help in power planning for the aggregators and energy saving for the consumers.

Very limited work exist in literature that work on the low sampled power measurements [8]–[11]. A knowledge based approach is presented in [8] which extracts spatial and temporal characteristics from the 15 minute sampled aggregate power data to classify loads broadly into different loads categories. Another work describes a Discriminative Sparse Coding (DDSC) method that uses dictionaries to learn the load characteristics [9]. Here, load disaggregation is carried out using hourly sampled power measurements. But, the load detection is poor since it is biased by the instances of load operation used during training the dictionaries. This method requires a lot of training data to learn all the instances of load operation which is practically not possible. Recently, few researchers have explored graph based methods utilizing Graph Signal Processing (GSP) techniques for carrying disaggregation [10], [12]. These methods are based on the assumption that the individual load signals are piece-wise smooth with respect to the graph created using aggregate measurements. A graph regularization based approach is presented in [12] for detecting loads from 1 minute sampled power measurements. This work is extended in [10], where graph signal smoothness based label propagation is utilized for both load identification and consumption estimation using 15 minutes aggregate power data.

Load disaggregation comprises of both load identification and consumption estimation from the aggregate power signal. In order to produce good disaggregation results, the algorithms should be good both at load detection and load operational waveform reconstruction, from which consumption can be estimated. Learning a good representation of loads is very much necessary to disaggregate loads in the low sampled case. Since the representation learning techniques learn the load characteristics, they are able to detect the loads more accurately [9], [11]. The graph based methods, on the other hand, are able to pick up the load operation times with good accuracy [10], [12]. A hybrid of these two techniques utilizing the dictionary representation (DLwG) is shown to perform well both at load detection and consumption estimation [11]. Transform Learning (TL) is another technique used for learning representation from the data. In literature, it has been used in various domains and has shown to produce state-of-theart results in a lot of applications [13]–[15]. Compared to dictionary, transforms are known to have superior performance with less computational complexity as brought out in [16]–[18].

Owing to the mentioned advantages, systematically combining transform representation of the loads with the graph approach for accurately extracting the load timing operation, emerges as a sensible novel proposal to carry out low-sampling electrical load disaggregation. This approach, referred to as TLwG, is analytically formulated in this paper and the necessary algorithmic steps are derived. The approach is tested with 15 and 30 minutes sampled aggregate power measurements using simulated and real dataset. Comparisons with the state-of-art dictionary counterpart (DLwG) [11], DDSC [9] and the GSP technique [10] showed that the proposed technique performs better than these methods in some of the cases; no appreciable degradation in others.

Towards providing the necessary details of the method, the paper is organized as follows. Section 2 presents the details of the proposed technique with the optimization formulation followed by the derivation of the closed form updates of the parameters. Section 3 provides the experimental results and discussion. Finally, Section 4 concludes the work.

II. Proposed Methodology

The proposed TLwG method involves an analytical formulation combining transform based representation learning and graph signal smoothness to arrive at robust disaggregation results. This method has a training phase where, transforms are learnt for the different loads using the load-specific data. In the test phase, the loads are estimated in an iterative manner by exploiting the graph signal smoothness property. The details of the two phases are presented in the subsequent sections.

A. Training Phase

Let the aggregate power measurements be given as $\bar{\boldsymbol{X}} \in \mathbb{R}^{T\times N}$, where N are the number of days and T are the measurements taken in a day. It is split into training and test data $\bar{\boldsymbol{X}} = [\bar{\boldsymbol{X}}_{train}|\bar{\boldsymbol{X}}_{test}]$. In the training phase, the individual load waveforms given by $\boldsymbol{X_m} \in \mathbb{R}^{T\times n}$ are known, where $n < N$ for the different loads of interest $m = 1, ..., M$. The aggregate training power data can be expressed as:

$$\bar{\boldsymbol{X}}_{train} = \sum_{m=1}^{M} \boldsymbol{X_m} + \boldsymbol{B} \tag{1}$$

where $\boldsymbol{B}$ includes other low power base loads not considered in M.

Individual load representations are learnt in form of transform $\boldsymbol{T_m}$ utilizing the load-specific power data $\boldsymbol{X_m}$ with the following formulation:

$$\boldsymbol{T_m X_m} = \boldsymbol{Z_m} \tag{2}$$

where $\boldsymbol{T_m} \in \mathbb{R}^{K\times T}$ is the data-driven transform that is learnt with K atoms and $\boldsymbol{Z_m}^{train} \in \mathbb{R}^{K\times n}$ are the transform coefficients. The sparse representation of the loads are learnt by enforcing sparsity on the coefficients $\boldsymbol{Z_m}^{train}$ using the following formulation [16]:

$$\min_{\boldsymbol{T_m}, \boldsymbol{Z_m}^{train}} \|\boldsymbol{T_m X_m} - \boldsymbol{Z_m}^{train}\|_F^2 + \lambda(\|\boldsymbol{T_m}\|_F^2 - \text{log det } \boldsymbol{T_m}) + \mu\|\boldsymbol{Z_m}^{train}\|_0 \tag{3}$$

where, the second term controls the condition number and prevents trivial solutions of $\boldsymbol{T_m}$ (see [14] for more details) and the third term involves the ℓ_0 norm that enforces sparsity on the learnt coefficients $\boldsymbol{Z_m}^{train}$. The closed form updates for $\boldsymbol{T_m}$ and $\boldsymbol{Z_m}^{train}$ is obtained by using alternate minimization following the approach given in [14]. The update for $\boldsymbol{Z_m}^{train}$ is expressed as:

$$\boldsymbol{Z_m}^{train} = (\text{abs}(\boldsymbol{T_m X_m}) \geq \mu) \odot \boldsymbol{T_m X_m} \tag{4}$$

where the term in the bracket is hard thresholded against the value μ and '$\odot$' denotes the element-wise product. The transform update is obtained by solving:

$$\boldsymbol{T_m} \leftarrow \min_{\boldsymbol{T_m}} \|\boldsymbol{T_m X_m} - \boldsymbol{Z_m}^{train}\|_F^2 + \lambda(\|\boldsymbol{T_m}\|_F^2 - \text{log det } \boldsymbol{T_m}). \tag{5}$$

Cholesky decomposition followed singular value decomposition is used to obtain the update for $\boldsymbol{T_m}$, again, following [14]. The main steps are given below:

$$\boldsymbol{X_m X_m}^T + \lambda \boldsymbol{I} = \boldsymbol{L L}^T \tag{6}$$

$$\boldsymbol{L}^{-1}\boldsymbol{X_m}(\boldsymbol{Z_m}^{train})^T = \boldsymbol{U S V}^T. \tag{7}$$

The requisite transform update is obtained as:

$$\boldsymbol{T_m} = 0.5\boldsymbol{V}(\boldsymbol{S} + (\boldsymbol{S}^2 + 2\lambda \boldsymbol{I})^{1/2})\boldsymbol{U}^T\boldsymbol{L}^{-1}. \tag{8}$$

The transforms learnt for the different loads are utilized for estimating the loads in the test phase.

B. Test Phase

In this phase, the transform coefficients $\boldsymbol{Z_m}^{test}$ are estimated for each of the loads using the aggregate test data $\bar{\boldsymbol{X}}_{test} \in \mathbb{R}^{T\times(N-n)}$ and the transforms learnt in the training phase. The coefficients represent the load activations and together with the respective transform are used to reconstruct the load waveform. In addition to load waveform reconstruction, it is essential to accurately pick up the load operation timing. Here, the timing information of load operation is identified using few concepts of GSP, where the activations from the training phase are propagated to the appropriate instant in the test phase utilizing the graph framework. A graph is constructed using aggregate power measurements to capture the correlation between the power measurements. The transform coefficients $\boldsymbol{Z_m}$ are treated as the graph signals where each column of $\boldsymbol{Z_m}$ resides on the corresponding node of the underlying graph. This is a reasonable representation as similar changes in the aggregate power measurements most likely correspond to the same load operation.

Using the aggregate power measurements $\bar{\boldsymbol{X}}$, a graph of N nodes is constructed with T samples residing on each node. The popular thresholded Gaussian kernel weighting function is used to estimate the weight matrix which is given as:

$$\boldsymbol{W}(i,j) = exp(-|dist(i,j)|^2/\sigma^2) \tag{9}$$

where σ is a scaling factor and $dist(i,j)$ is the distance between the values residing at nodes i and j. To handle simultaneous operation of multiple loads, the difference of the adjacent aggregate power measurements ($\boldsymbol{\Delta p}$) are considered for graph construction. Euclidean distance measure is adopted to compute the weights of the graph. Using the weight matrix $\boldsymbol{W}$, the graph Laplacian $\boldsymbol{L}$ is obtained as [19]: $\boldsymbol{L} = \boldsymbol{D}^G - \boldsymbol{W}$, where, $\boldsymbol{D}^G$ is the degree matrix of the graph.

Since $\boldsymbol{Z_m}$ is partially known (${\boldsymbol{Z_m}}^{train}$ being available), the graph signal corresponding to the test phase can be computed by minimizing the total variation or increasing the smoothness of the signal with respect to the underlying graph. In this way, the signal values gets propagated through the graph from the training phase to the test phase utilizing the weight information which reflects the similarity of the power measurements. This results in the following joint optimization formulation:

$$\min_{\boldsymbol{Z_m}} \|\bar{\boldsymbol{X}} - \sum_{\boldsymbol{m \in M}} (\boldsymbol{T_m})^{\dagger} \boldsymbol{Z_m}\|_F^2 + \beta \sum_{m \in M} tr(\boldsymbol{Z_m L_m Z_m}^T) \quad (10)$$

where $\bar{\boldsymbol{X}} = [\bar{\boldsymbol{X}}_{train} | \bar{\boldsymbol{X}}_{test}]$ and $\boldsymbol{Z_m} = [{\boldsymbol{Z_m}}^{train} | {\boldsymbol{Z_m}}^{test}]$ are the transform coefficients for load m. Here, $\dagger$ denotes pseudo inverse and $tr()$ is the trace of the matrix. $\boldsymbol{L_m}$ is the combinatorial graph Laplacian of the m-th load. The first term in (10) is the fidelity term which computes the coefficients using the learnt transforms such that the error is minimized. The second term is the graph smoothness term which ensures that estimated coefficients are smooth with respect to the underlying graph.

As mentioned earlier, the coefficients ${\boldsymbol{Z_m}}^{train}$ are known from the training data, hence, solving for $\boldsymbol{Z_m}$ reduces to estimating ${\boldsymbol{Z_m}}^{test}$. Since different $\boldsymbol{L_m}s$ are *not* available in the test phase, (10) is re-formulated to obtain each of the $\boldsymbol{L_m}$ greedily, starting from the aggregate $\bar{\boldsymbol{X}}$. The coefficients ${\boldsymbol{Z_m}}^{test}$ are obtained for each load one at a time in an iterative manner till all loads are disaggregated. This leads to the following formulation:

$$\min_{{\boldsymbol{Z_m}}^{test}} \|\bar{\boldsymbol{X}}_{test} - (\boldsymbol{T_m})^{\dagger} {\boldsymbol{Z_m}}^{test}\|_F^2 + \beta\ tr(\boldsymbol{Z_m L_m Z_m}^T). \quad (11)$$

Minimizing the graph smoothness term in (11) results in the following expression [11]:

$$\min_{\boldsymbol{Z_m}} tr(\boldsymbol{Z_m L_m Z_m}^T) = \min_{{\boldsymbol{Z_m}}^{test}} tr(2{\boldsymbol{Z_m}}^{test} \boldsymbol{L_m}(n+1:N, 1:n)$$
$$.({\boldsymbol{Z_m}}^{train})^T + {\boldsymbol{Z_m}}^{test} \boldsymbol{L_m}(n+1:N, n+1:N)({\boldsymbol{Z_m}}^{test})^T). \quad (12)$$

The solution for ${\boldsymbol{Z_m}}^{test}$ is obtained by taking a derivative of (11) with respect to ${\boldsymbol{Z_m}}^{test}$ and equating it to zero. This results into a Sylvester equation [20] of the form:

$$(\boldsymbol{T_m})^{\dagger T} (\boldsymbol{T_m})^{\dagger} {\boldsymbol{Z_m}}^{test} + \beta {\boldsymbol{Z_m}}^{test} \boldsymbol{L_m}(n+1:N, n+1:N)$$
$$= (\boldsymbol{T_m})^{\dagger T} \bar{\boldsymbol{X}}_{test} - \beta {\boldsymbol{Z_m}}^{train} \boldsymbol{L_m}(n+1:N, 1:n)^T \quad (13)$$

from which the coefficients ${\boldsymbol{Z_m}}^{test}$ are estimated for load m; these coefficients together with the $\boldsymbol{T_m}$ (learnt during the training phase) leads to the disaggregated waveform of load m in the test phase as follows:

$$\hat{\boldsymbol{X}}_{\boldsymbol{m}} = (\boldsymbol{T_m})^{\dagger} {\boldsymbol{Z_m}}^{test}. \quad (14)$$

In the direction of resolving superpositions effectively, disaggregation is carried out in an iterative manner starting from the highest power consuming load (see also [11]). Once the load is estimated its contribution is removed from both the total aggregate $\bar{\boldsymbol{X}}$ and the test aggregate power measurements $\bar{\boldsymbol{X}}_{test}$ before considering the next load. A new graph is created using the modified $\bar{\boldsymbol{X}}$ for the next load and this procedure repeats till all the loads are disaggregated. Fig. 1 presents the flow diagram of the proposed TLwG methodology for load disaggregation with the corresponding pseudo code given in Algorithm 1.

Algorithm 1 Transform Learning with Graph Smoothness (TLwG)

Input: Aggregate Power measurements $\bar{\boldsymbol{X}} = [\bar{\boldsymbol{X}}_{train} | \bar{\boldsymbol{X}}_{test}]$, Load-specific power data, $\boldsymbol{X_m}$ for $m = 1, .., M$ loads, K (size of transforms (atoms)), parameters (λ, μ, β) and maximum number of iterations $Maxitr$
Output: Learnt transforms $\boldsymbol{T_1}$,.., $\boldsymbol{T_M}$ and estimated loads $\hat{\boldsymbol{X}}_{\boldsymbol{1}}$, .., $\hat{\boldsymbol{X}}_{\boldsymbol{M}}$
Initialization: Set $\boldsymbol{T_1}$,.., $\boldsymbol{T_M}$ to random matrix with real numbers between 0 and 1 drawn from a uniform distribution. ${\boldsymbol{Z_1}}^{train}$ = $\boldsymbol{T_1 X_1}$,.., ${\boldsymbol{Z_M}}^{train}$ = $\boldsymbol{T_M X_M}$ and iteration $i = 1$

```
procedure TRAINING PHASE
    Set m = 1
    while m ≤ M do
    loop: Repeat until convergence (or fixed number of iterations Maxitr)
    Z_mi^train ← update using (4) with T_mi
    T_mi ← update using (6)-(8)
    i ← i + 1
    if ||T_mi − T_m(i−1)||_F < Tol or i == Maxitr then
        m ← m + 1
    else go to loop
    end if
    end while
procedure TEST PHASE
    Set m = 1
    while m ≤ M do
    Construct graph L_m using X̄ in (9)
    Compute Z_m^test using (13)
    Estimate the load X̂_m using (14)
    X̄ ← X̄ − [X_m | X̂_m]
    X̄_test = X̄_test − X̂_m
    m ← m + 1
    end while
```

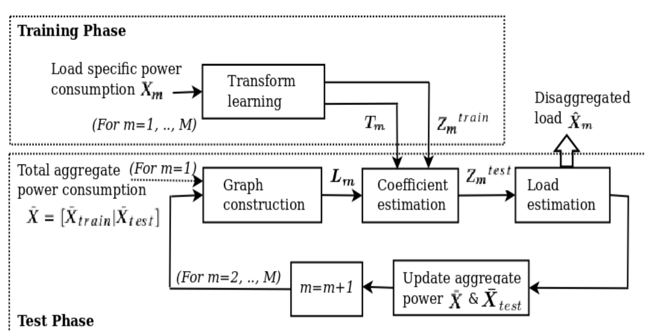

Fig. 1. Flow Diagram of the Proposed TLwG Method for Load Disaggregation

III. Results and Discussion

The proposed methodology for load disaggregation is evaluated using both simulated and real datasets for low-sampled power measurements. For the simulated data, Home Energy

Simulator (HES) [21] is used to generate aggregate power measurements at every 15 minutes intervals. The real dataset makes use of the public UMass Smart Home Dataset [22] with power measurements sampled at 30 minutes intervals. Results of the proposed technique is compared against the Dictionary counterpart (DLwG) [11], the related DDSC [9] and the GSP technique [10]. Instead of trying to disaggregate all possible loads, which gets too challenging with low sampled power measurements, the disaggregation is focused only on the high power consuming loads with distinct power levels. The popular F_1 score is utilized for load identification. Load consumption estimation is evaluated using the average percentage accuracy ($\%Acc$) defined as:

$$\%Acc = \frac{1}{N}\sum_{i=1}^{N}\left[1 - \frac{(|\boldsymbol{X}_i^{act} - \boldsymbol{X}_i^{pred}|)}{\boldsymbol{X}_i^{act}}\right] \times 100 \qquad (15)$$

where $\boldsymbol{X}^{act}$ is the actual and $\boldsymbol{X}^{pred}$ is the estimated power consumption of the load.

The transforms learnt for the different loads make use of the same atom size K and sparsity threshold μ and are appropriately tuned for both the datasets. The dictionary method utilizes the K-SVD toolbox in matlab [23] to learn the dictionaries for the respective loads with K = 100 atoms and sparsity threshold τ = 20. Graph is constructed using the normalized $\boldsymbol{\Delta p}$ values of the aggregate power measurements with $\sigma = 0.5$ for both the datasets. To capture the load operation timings more effectively, higher weightage is given to the graph smoothness term while estimating the loads. Accordingly, the value of hyperparameter β is tuned for both the datasets. The following section presents the results obtained with the simulated and real dataset for load disaggregation.

A. Results with Simulated Data

As mentioned earlier, Home Energy Simulator is used to simulate the power consumption profile of the household. The simulator comprises of load traces from the public REDD [24] and tracebase datasets [25] sampled at 10 seconds. The data is downsampled to 15 minutes intervals by taking an average over a 15 minutes window. Data for 300 days are generated for the commonly used loads namely, Dryer, Dishwasher, Geyser, Refrigerator and other low power base loads. Training is carried out using 190 days data where, transforms (with $K = 96$) are learnt for the loads of interest. The remaining 110 days are used for testing.

To compare with the results existing in [11], we considered the same three loads: Dryer, Dishwasher and Refrigerator. Fig. 2 presents the load disaggregation results obtained with the proposed TLwG technique along with ground truth. The disaggregation results for both load identification and load consumption estimation obtained with the different methods are summarized in Table 1. It can be seen, for Dryer, both the GSP and DLwG techniques have good load identification accuracy. The consumption estimate is good for both the proposed transform technique and the dictionary counterpart with the latter being more superior. For other loads, Dishwasher and Refrigerator, the transform technique performs the best, both

TABLE I
COMPARATIVE RESULTS WITH HES DATASET

Loads	**Dryer**		**Dishwasher**		**Refrigerator**	
Algorithms	F_1	$\%Acc$	F_1	$\%Acc$	F_1	$\%Acc$
TLwG	0.85	84.02	**0.82**	**82.04**	1	**86.25**
DLwG [11]	**0.91**	**86.47**	0.81	76.60	1	73.48
DDSC [9]	0.43	51.05	0.39	58.48	1	40.51
GSP [10]	0.90	70.59	0.80	67.04	1	63.43

TABLE II
COMPARATIVE RESULTS WITH UMASS DATASET

Loads	**Dryer**		**AC**		**Furnace**	
Algorithms	F_1	$\%Acc$	F_1	$\%Acc$	F_1	$\%Acc$
TLwG	**0.76**	**83.69**	**0.80**	73.19	1	62.18
DLwG [11]	**0.76**	82.83	0.76	**76.61**	1	**71.78**
GSP [10]	0.74	44.15	0.67	65.79	1	64.69

in terms of F_1 score and $\%Acc$. Refrigerator has F_1 score of 1 as it was continuously ON and was correctly detected by all the methods. It can be seen, the GSP-based technique is good at load detection than load waveform reconstruction. On the other hand, the representation learning techniques based on transform and dictionaries have better load reconstruction capability. Incorporating the graph structure with transform or dictionary representation (TLwG and DLwG) results in both improved load detection and consumption estimation.

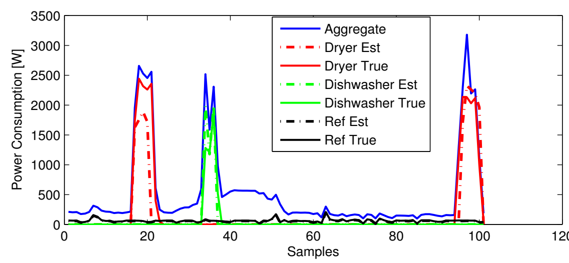

Fig. 2. Disaggregation Results with HES Data

B. Results with Real Data

The publicly available UMass Smart Home Dataset is considered here which contains power data collected from three real homes for three years. The load and the aggregate power data are sampled at 30 minutes intervals. For this dataset, one year data i.e. 365 days is taken from training and testing is carried out for the next 100 days. The aggregate power majorly comprises of Dryer, AC, Furnace, Microwave, Refrigerator and other low power base loads.

Disaggregation is carried out for Dryer, AC and Furnace. Fig. 3 presents the disaggregation results for a day using the TLwG (with $K = 48$) technique along with ground truth. Table 2 summarizes the results obtained with all the methods for load detection and estimation. It can be seen that both TLwG and DLwG techniques have better performance than the GSP technique both in terms of F_1 score and $\%Acc$. Contrary to the previous 15 minutes case, TLwG has better load detection accuracy than load estimation. Furnace has F_1 score of 1 as it was continuously ON and was correctly detected by all the methods. Comparison with the DDSC method is not given as its performance was very low (see also the observations related to DDSC in [11]).

Since the proposed TLwG method disaggregates loads in an iterative manner, false positives and false negatives of

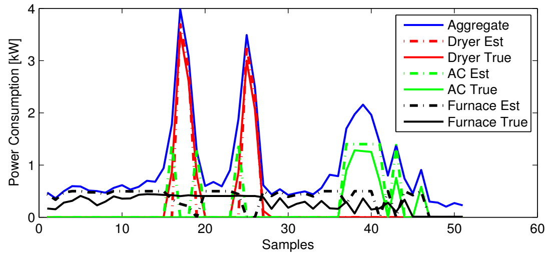

Fig. 3. Disaggregation Results with Umass Data

load estimation have a significant impact on the estimation of next load being considered. Thus, threshold based post-processing similar to the DLwG method is adopted to limit the error propagation. It can be seen from Tables 1 & 2 that, as the sampling interval of the power measurements increases from 15 to 30 minutes, the load detection accuracy reduces. This is due to the fact, at such long intervals, many loads exhibit similar operational behavior making it difficult to disaggregate them. Although the load detection accuracy decreases, the consumption estimates for the loads using both TLwG and DLwG techniques are still good. This emphasizes the importance of learning a good representation of the loads for improved disaggregation results. It can be observed that the proposed method provides marginal performance improvement over the dictionary method (DLwG) in some cases, but is more computationally efficient (again, see [16]–[18] for more details on computational aspect comparisons). In summary, TLwG provides another way of learning the data representation that has computational advantage over dictionaries without any significant performance degradation.

IV. Conclusion

Towards combining right representation with timing extraction to address the source separation problems with complex superpositions, this paper presented a method employing transform based representation learning combined with graph signal processing to estimate individual load waveforms of interest. The initial results obtained are very promising. Even though the application focus is a challenging low-sampled electrical load disaggregation or non-intrusive load monitoring, the method can be utilized in other source separation scenarios, which are becoming important in present day applications due to ubiquitous use of sensors.

References

[1] Yi Wang, Qixin Chen, Tao Hong, and Chongqing Kang, "Review of smart meter data analytics: Applications, methodologies, and challenges," *CoRR*, vol. abs/1802.04117, 2018.

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

[3] O. P. Patri, A. V. Panangadan, C. Chelmis, and V. K. Prasanna, "Extracting discriminative features for event-based electricity disaggregation," in *2014 IEEE Conference on Technologies for Sustainability (SusTech)*, July 2014, pp. 232–238.

[4] Oliver Parson, Siddhartha Ghosh, Mark Weal, and Alex Rogers, "Using hidden markov models for iterative non-intrusive appliance monitoring," in *Neural Information Processing Systems workshop on Machine Learning for Sustainability*, December 2011.

[5] X. Wang, J. Wang, D. Shi, and M. E. Khodayar, "A factorial hidden markov model for energy disaggregation based on human behavior analysis," in *IEEE Power Energy Society General Meeting (PESGM)*, Aug 2018, pp. 1–5.

[6] Hyungsul Kim, Manish Marwah, Martin Arlitt, Geoff Lyon, and Jiawei Han, "Unsupervised disaggregation of low frequency power measurements," April 2011, vol. 11, pp. 747–758.

[7] K. Carrie Armel, Abhay Gupta, Gireesh Shrimali, and Adrian Albert, "Is disaggregation the holy grail of energy efficiency? the case of electricity," *Energy Policy*, vol. 52, pp. 213 – 234, 2013, Special Section: Transition Pathways to a Low Carbon Economy.

[8] Chandra M.G. Srinivasarengan K., Goutam Y.G., "A knowledge based approach for disaggregation of low frequency consumption data," *Thampi S., Gelbukh A., Mukhopadhyay J. (eds) Advances in Signal Processing and Intelligent Recognition Systems. Advances in Intelligent Systems and Computing, Springer*, vol. 264, 2014.

[9] J. Zico Kolter, Siddarth Batra, and Andrew Y. Ng, "Energy disaggregation via discriminative sparse coding," in *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*, USA, 2010, NIPS'10, pp. 1153–1161, Curran Associates Inc.

[10] K. Kumar and M. G. Chandra, "An intuitive explanation of graph signal processing-based electrical load disaggregation," in *IEEE 13th International Colloquium on Signal Processing its Applications (CSPA)*, March 2017, pp. 100–105.

[11] Kriti Kumar, M. Girish Chandra, A. Anil Kumar, and Naveen Kumar Thokala, "Low sampling rate electrical load disaggregation using dictionary representation and graph signal smoothness," in *Proceedings of the Tenth ACM International Conference on Future Energy Systems (ACM e-Energy '19)*, New York, NY, USA, 2019, pp. 47–51.

[12] V. Stankovic, J. Liao, and L. Stankovic, "A graph-based signal processing approach for low-rate energy disaggregation," in *IEEE Symposium on Computational Intelligence for Engineering Solutions (CIES)*, Dec 2014, pp. 81–87.

[13] K. Kumar, A. Majumdar, M. G. Chandra, and A. Anil Kumar, "Transform learning based functionapproximation for regression and forecasting," in *4th ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data*, September 2019.

[14] J. Maggu and A. Majumdar, "Kernel transform learning," *Pattern Recognition Letters*, vol. 98, pp. 117 – 122, 2017.

[15] S. Ravishankar and Y. Bresler, "Learning doubly sparse transforms for images," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 4598–4612, Dec 2013.

[16] S. Ravishankar and Y. Bresler, "Learning sparsifying transforms," *IEEE Transactions on Signal Processing*, vol. 61, no. 5, pp. 1072–1086, March 2013.

[17] S. Ravishankar, B. Wen, and Y. Bresler, "Online sparsifying transform learning part i: Algorithms," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 625–636, June 2015.

[18] S. Ravishankar and Y. Bresler, "Learning overcomplete sparsifying transforms for signal processing," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 3088–3092.

[19] D. I Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, May 2013.

[20] Rajendra Bhatia, *Matrix Analysis*, vol. 169, Springer, 1997.

[21] Krishnan Srinivasarengan, Y. G. Goutam, and M. Girish Chandra, "Home energy simulation for non-intrusive load monitoring applications," in *Proceedings of International Workshop on Engineering Simulations for Cyber-Physical Systems (ACM ES4CPS '14)*, New York, NY, USA, 2007, pp. 9:9–9:12.

[22] Sean Barker, Aditya Mishra, David Irwin, Emmanuel Cecchet, Prashant Shenoy, and Jeannie Albrecht, "Smart*: An open data set and tools for enabling research in sustainable homes," .

[23] M. Aharon, M. Elad, and A. Bruckstein, "lt;tex gt;rmk lt;/tex gt;-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, Nov 2006.

[24] J. Zico Kolter and Matthew J. Johnson, "REDD: A Public Data Set for Energy Disaggregation Research," in *SustKDD Workshop on Data Mining Applications in Sustainability*, 2011.

[25] A. Reinhardt, P. Baumann, D. Burgstahler, M. Hollick, H. Chonov, M. Werner, and R. Steinmetz, "On the accuracy of appliance identification based on distributed load metering data," in *2nd IFIP Conference on Sustainable Internet and ICT for Sustainability (SustainIT 2012)*, Oct. 2012, vol. 00, pp. 1–9.