Unsupervised disaggregation of appliances using aggregated consumption data

Hugo Gonçalves[1]
[1] Electrical and Computer Engineering
Carnegie-Mellon University
Pittsburgh, PA 15213
hgoncalv@andrew.cmu.edu

Adrian Ocneanu[2]
[2]Civil and Environmental Engineering
Carnegie-Mellon University
Pittsburgh, PA 15213
aocneanu@andrew.cmu.edu

Mario Bergés[2]
[2]Civil and Environmental Engineering
Carnegie-Mellon University
Pittsburgh, PA 15213
mberges@andrew.cmu.edu

Abstract

Non-Intrusive Load Monitoring (NILM) is a technique that determines the electrical load composition of a household through a single point of measurement at the main power feed. In contrast with the majority of the existing approaches to solve this problem which require training, here we explore an unsupervised approach to determine the number of appliances in the household, their power consumption and state, at any given moment. We attempt to achieve this without using any *a priori* information on the number and type of appliances. Our approach is to first create clusters of steady-state changes and then employ a matching pursuit algorithm to reconstruct the original power signals using the clusters that were found as the sources in a linear blind source separation strategy. Changes in steady-state, sometimes referred to as events, are characterized by their change in real and reactive power (P and Q). Ultimately, the results may be applied to other features in an attempt to improve the separation between clusters.

The preliminary results point toward a mixed scenario: large appliances (roughly above 400W) were easily identified, but the small appliances typically clustered together and were difficult to separate. We conclude that the errors occur during clustering which indicates that, in order to increase the purity of the clusters, perhaps other features could be used.

Author keywords

Non-intrusive load monitoring, blind source separation, unsupervised consumption disaggregation, energy consumption, clustering

1. INTRODUCTION

Non-Intrusive Load Monitoring (NILM) is a technique that determines the load composition of a household through a single point of measurement, typically at the main power feed [9]. The current focus of NILM is the disaggregation of load states by means of supervised learning algorithms that use transition signatures. A transition signature is any set of features that describes a power consumption event (e.g., a turn ON or a turn OFF). This signature is used in a supervised learning approach to label new events [1].

The main stages and challenges of NILM are: (i) the detection of transitions and (ii) their classification. The first stage is normally done by means of statistical change detection [8] and ongoing research is aimed at improving its performance [11]. The second stage of a supervised approach is based on an existing library of labeled transitions with which any classification algorithm can work (e.g., k-Nearest Neighbors). However, obtaining the labels for the signatures is typically done manually, by a user, or through the use of sub-metering. In most cases, these methods are either impractical and/or expensive. There have also been efforts towards automating this training step [12].

In the approach presented here, we attempt to identify an unsupervised learning alternative for load disaggregation, using Blind Source Separation (BSS) [14]. The underlying motivation for such an approach is to explore its applicability to load disaggregation and reduce (or ideally remove) the need for training. We are interested in, eventually, applying it to different sets of features that can be extracted from aggregate voltage and/or current signals. However, for our initial studies, we chose to use some of the most studied ones: the real and reactive powers. Before we commit to any supplementary research efforts, we tried to assess the usefulness of such an approach.

1.1 Signal and its representation

We assume that the observation dataset $O \in R^{2xt}$, where 2 represents the real (P) and reactive (Q) power and t is the sample number, is composed of a linear combination of sources $S \in R^{2xs}$, one for each appliance cluster, weighted by the weight matrix $W \in R^{sxt}$.

$$SW = O$$

$$\begin{bmatrix} S_P^1 & \dots & S_P^s \\ S_Q^1 & \dots & S_Q^s \end{bmatrix} \times \begin{bmatrix} W_1^1 & \dots & W_1^t \\ \dots & \dots & \dots \\ W_s^1 & \dots & W_s^t \end{bmatrix} = \begin{bmatrix} P \\ Q \end{bmatrix} \quad (1)$$

Our goal is to iteratively explain $\boldsymbol{O}$ in terms of a 'best-fit' solution from an approximated weight matrix $\boldsymbol{W}$ and the source matrix $\boldsymbol{S}$.

In particular, the source matrix $\boldsymbol{S}$ is composed of the instantaneous power consumption of the appliances and the weight matrix should be filled with binary data, since we work under the assumption that appliances are either ON or OFF.

The signal $\boldsymbol{O}$, presented in **Error! Reference source not**

SustKDD 2011, August 2011, San Diego, CA, USA

found., was retrieved from an instrumented house, using a prototype system developed at Carnegie Mellon University [2]. The ground truth was obtained by sequentially actuating a selection of appliances manually (Table 1). A built-in event detector prompted a user to label the events. This dataset has a time duration of roughly 1.6 hours (with power and reactive power values computed at 60Hz). We focused on relatively simple appliances in terms of their transients and operations, as the focus of this work was to provide a judgment on whether the chosen features are suitable to separate the appliances blindly. These appliances are identified by their ID, in a database.

Table 1 Appliances with IDs

ID	Appliance name
68	Refrigerator (Cooling)
69	Microwave
70	Oven (Upper)
71	Oven (Lower)
72	Garbage Disposal
74	Breakfast Area Under Cabinet Lights
75	Breakfast Area In Cabinet Lights
76	Breakfast Area Hanging Lights
77	Breakfast Area Overhead Lights
78	Kitchen Hanging Light
80	Kitchen Under Cabinet Lights
81	Range Hood Fan
84	Range Hood Light
86	Kitchen Hallway Overhead Lights
155	Toaster
157	Coffee Machine
158	Refrigerator (Light)
182	Basement Stairs Light

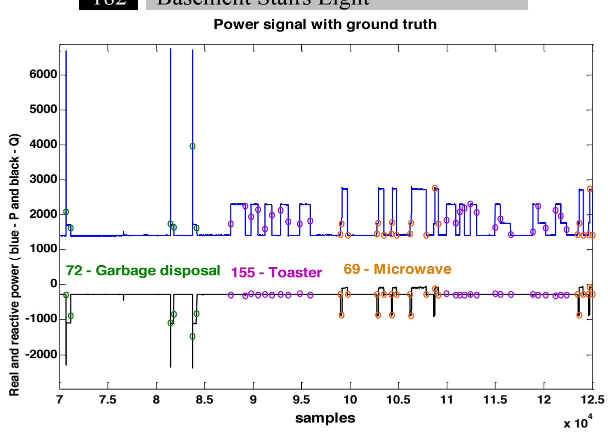

Figure 1 Subset of original signal with ground truth

2. METHODS

As can be seen in **Error! Reference source not found.**, the power signal is mainly composed of step-up/down events corresponding to the switching on and off of appliances. Because the signal is dominated by steady-state periods, the approach taken was to first detect the steady-state changes in the power signal, measure the average change in power (both active and reactive) and use these features to automatically cluster the events. The clustering procedure returns the number of possible clusters and their centroid. Finally, these centroids are provided to the source reconstruction algorithm to explain, at each moment, which sources are active.

We assume that each cluster corresponds to one appliance state-transition (i.e., changes in steady-state power caused by the transition of an appliance in one state to another). Furthermore, we assume that all appliances that have more than two states (e.g., *on* and *off*) will produce events that can be explained by two-state appliances.

We employ a simple change-of-mean detector, based on the ideas described in [9]. We consider a moving window that slides throughout the whole power signal and compares the new samples with the window mean.

2.1 Clustering procedure

The clustering procedure should determine automatically how many clusters are in the data (corresponding to the number of different appliance state-transitions) and the centroid of these clusters (corresponding to the average power change for those transitions). Two types of clustering methods were investigated: (a) clustering based on genetic algorithms and (b) hierarchical agglomerative clustering. The best results were obtained with genetic k-means.

2.1.1 Genetic and K-means Clustering Algorithm

The genetic algorithm used in this approach was based on [7] and works in an iterative and evolutionary manner. First, clustering solutions are randomly initialized; these are called chromosomes. The fitness function shown in eq. 2 is used to assess if a solution is viable or not. The viable solutions are kept and mutated and the rest are discarded. The viable chromosomes generate offsprings, and all of the population is evaluated, using the same metric. K-means is again applied to each offspring and both solutions (the parent and the offspring) are measured by a fitness function. The better result prevails, while the other is removed. The fitness function is an adaptation to the original CS measure presented in [3]:

$$CS = \frac{\sum_{k=1}^{K} \frac{1}{M_k} \sum_{X_i \in C_k} \max_{X_q \in C_k} d(X_i, X_q)}{\sum_{i=1}^{K} \min_{i \neq j} \hat{d}(m_i, m_j)} \tag{2}$$

where K is the number of clusters in the solution, M_k is the number of points in cluster k, X_i is an event, m_i stands for $(P_c, Q_c)_i$, that are the coordinates of cluster i on the PQ plane, $d(X_i, X_q)$ is the Euclidian distance between X_i and X_q, and

$$\hat{d}(X_i, X_q) = \frac{d(X_i, X_q)}{\sqrt{\frac{X_i + X_q}{2}}} \tag{3}$$

2.1.2 Hierarchical Agglomerative Clustering

Hierarchical clustering algorithms can be categorized as either top-down or bottom-up [4]. Bottom-up algorithms treat each data point as a singleton cluster at the outset and then successively merge pairs of clusters until all the clusters have been merged into a single cluster that contains all data points. Bottom-up hierarchical clustering is therefore called hierarchical agglomerative clustering (HAC). Merge operations in HAC are assumed to be monotonic, which means that at each step the best merge available is taken.

An important component of a clustering algorithm is the distance measure between data points. In this approach, we tested the following two:

1) Single Link [6] – two clusters having minimum distance between their closest data points are merged together.

$$D_{sl}(C_i, C_j) = \min_{x,y}\{d(x,y) \mid x \in C_i, y \in C_j\} \quad (4)$$

2) Complete Link [6] – At each step, the two clusters having minimum distances between the two farthest points are merged together.

$$D_{cl}(C_i, C_j) = \max_{x,y}\{d(x,y) \mid x \in C_i, y \in C_j\} \quad (5)$$

Agglomerative clustering results are typically represented in the form of a dendrogram (a tree structure, that is formed by decomposing data points into several levels of nested partitions). The final clusteres are obtained by cutting the dendrogram at the desired level (threshold).

2.1.3 *Clustering evaluation*

When ground truth is present, a sound method of performing clustering evaluation is purity. We use a slightly altered version of [6], by computing it for every individual cluster. We define purity as the number of elements in the most frequent class divided by the total number of elements in the cluster:

$$purity_j(\Omega, C) = \frac{1}{N_j}\sum_k \max_j |\omega_k \subset c_j| \quad (6)$$

where Ω is the set of all elements with for which the label is known, C is the set of all clusters created, ω_k is the subset with the highest frequency in each c_j cluster and N_j is the total number of elements in every cluster.

2.2 Source reconstruction - Matching pursuit (MP)

The centroids of the clusters correspond to the entries of our source matrix ***S***. The remaining step is to determine the entries of the weight matrix ***W*** which we do by using a matching pursuit algorithm that tries to match each event to one (or more) of the sources.

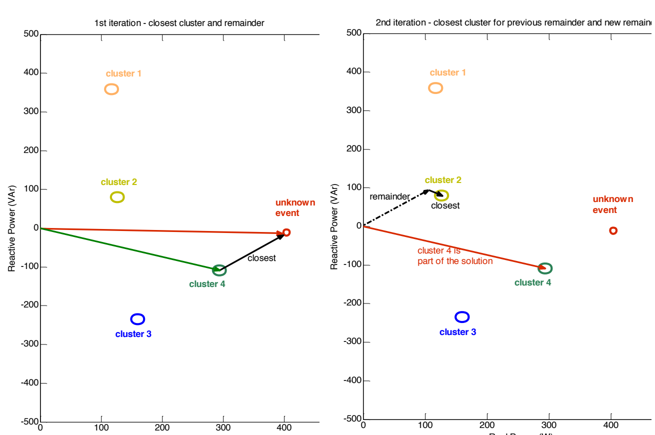

Figure 2 Iterations of matching pursuit on generic sources and a generic unknown event

This greedy algorithm is based on the idea that an event is a summation of sources, so it should 'match the most of the event with the smallest number of components' possible [13]. All of the entries of the weight matrix are initialized to zero. The algorithm works by processing each one of the detected events in the order in which they occur. Iteratively, it tries to match each event (X_i) to the closest source (m_i), in terms of Euclidean distance, and then subtracts this source from it ($X_i - m_i$). This process is shown in Figure 2 for a single event. For the algorithm to converge, we introduced a 'drain' source located at the origin (0,0). Put differently, the algorithm stops when a residual vector is closer to 'noise' (the drain) than to any sources present.

There are two main points to be made here, representing adaptations made to the original matching pursuit algorithm:

i) The matching is performed in terms of Euclidian distance ($[P_{obs}, Q_{obs}]$-$[P_{closest}, Q_{closest}]$) instead of using dot products.

ii) Once a source is matched to an event, its corresponding entry in the weight matrix is changed to 1 from the column corresponding to the time of occurrence of this event, until the column corresponding to the next event.

2.3 Coincidence detection

Some of the detected events may be the overlap of two or more coincident events, which we designate by 'mixing' events. Therefore, if the matching pursuit algorithm is employed without taking this into consideration, the resulting weight matrix may contain two consecutive turn ONs or OFFs registered for the same appliance, which we know is not possible. To detect this type of situation, a post-processing was performed on the results of the clustering. **Error! Reference source not found.** depicts this procedure.

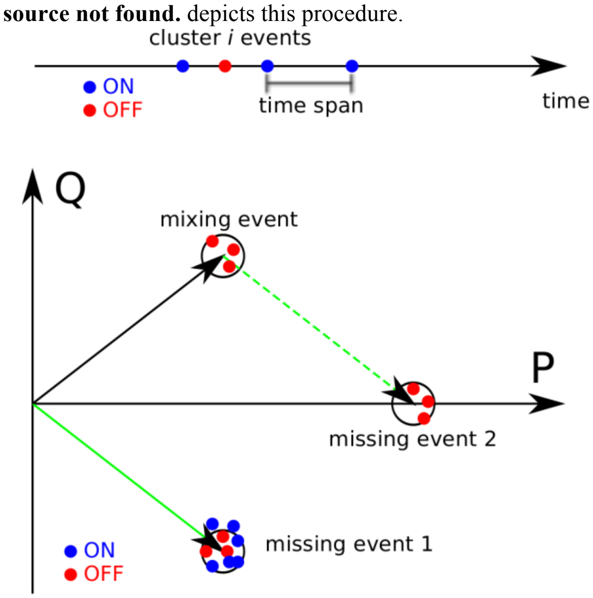

Figure 3 Coincidence detector

Based on the assumption that a cluster should contain a similar number of turn OFFs as turn ONs, as we overlapped the turn ON and turn OFF events by inverting the signs of the latter. Thus the 2nd quadrant overlaps on the 4th and the 3rd quadrant on the 1st, respectively. This applies to any of the representations of any cluster.

We consider for each cluster an asymmetry factor defined by

$$af(i) = (n_{i(on)} - n_{i(off)})/(n_{i(on)} + n_{i(off)}) \quad (7)$$

where $n_{i(on)}$ and $n_{i(off)}$ are the number of ON and OFF events, respectively.

The probability that a point belongs to cluster m is defined as $P_k(m)$, based on the distance to its center and assuming the cluster has a Gaussian distribution. A metric that measures how probable X_j is a mixing event is defined as

$$M(i,j,k,m) = \frac{af(i) \cdot af(m) \cdot P_k(m)}{P_j} \quad (8)$$

Values of M higher than M_{min} are considered as mixing/missing events and the mixing event is split into the corresponding missing events.

3. RESULTS

3.1 Event Detector

The confusion matrix in Table 2 shows the performance of the event detector. To determine correct detections, a maximum temporal difference of 1s (60 samples) was allowed between a detected event and a labeled sample.

Our dataset contains 296 labeled events. The event detector detected 299 events; the differences are indicated in Table 2.

Table 2 Confusion Matrix

	No ground truth	Ground truth
No event detected	354701 – true negative	2 – false negative
Event detected	5 – false positive	294 – true positive

These differences exist primarily for two reasons:

i) Not all events in the dataset were labeled (the manual labeling process was not perfect) resulting in false positives which are in fact correct detections (true positives).

ii) We opted for a very simple event detector, which inevitably misses some of the actual events. Other more sophisticated event detectors, like the one described in [11], may eliminate these errors.

3.2 Clustering

3.2.1. Clusters

Based on the 294 matching events and ground truth points, a P/Q representation of each event was created in Figure 4.

Regarding the natural clustering of these events, there are a few appliances that have a multi-stage consumption schema, the most obvious one being 69 (Microwave) and 155 (Toaster).

The nature of our ground truth gathering recorded the same appliance on its various stages of functioning, without discriminating between its many consumption levels.

A possible issue with later clustering may occur with appliances that are very close in terms of dP and dQ, namely at low power levels (less than 400W, presented in Figure 5) as witnessed in Figure 6, and discussed later. This clearly contrasts with the supervised approach, where similar events' classes can be well defined and separated [9].

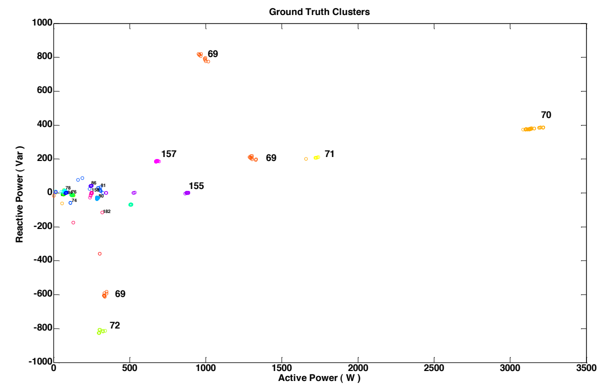

Figure 4 Ground truth clusters

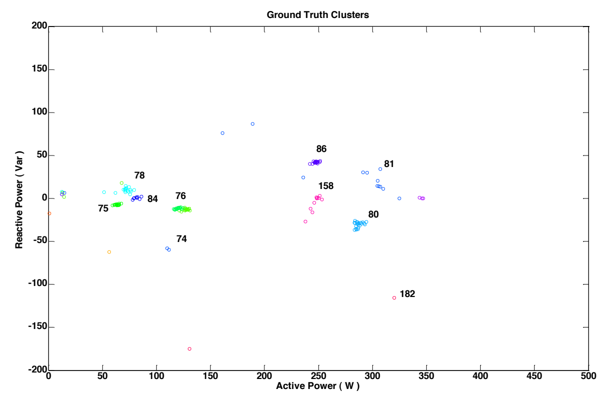

Figure 5 Detail of lower level consumption appliances

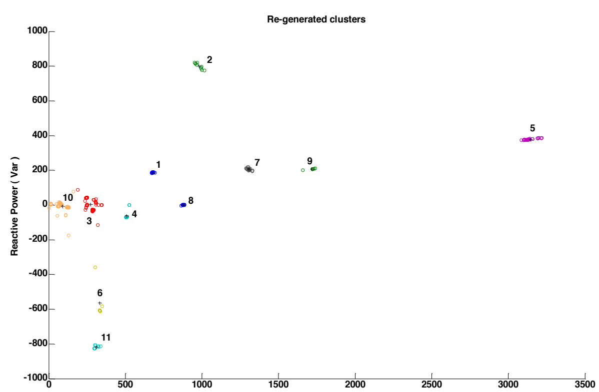

Figure 6 Generated clusters after coincidence detection

3.2.2. Constructed clusters

By plotting all of the events detected that had ground truth, we obtained the clusters presented in Figure 6.

As expected, since it's an inherent result of the clustering technique, we noted that a multistate appliance may end-up being split into two or more clusters: note appliance 69 (Microwave) or 155 (Toaster), as can examined in Figure 6.

The reverse is also true, since we also noted two clusters that each enclosed several appliances, as it can be seen in the same Figure 6, when compared to Figure 5.

3.2.3 *Clustering evaluation*

Based on the metric defined in section 2.2.3 and considering the final results in terms of clustering from Figure 6, the results are presented in Table 3.

Table 3 Purity metric results

	Cluster number	Predominant appliance	Purity of cluster
Clusters	1	157	1
	2	69	1
	3	80	0.32
	4	77	0.9
	5	70	1
	6	69	0.83
	7	69	1
	8	155	1
	9	71	0.85
	10	74	0.21
	11	72	1

3.3 Coincidence detection

Appliance 69 (Microwave) presents a multistage behavior, and it was modeled by the clustering algorithm as three clusters (see clusters 4, 8 and 10 in Figure 5). The coincidence detector worked by leveling the number of turn ON events: 4 extra turn ONs were eliminated from cluster 4 and in return they were explained as 4 missing turn ONs of cluster 8 and 4 missing OFFs of cluster 10.
After the coincidence detection relocated the 4 events, new clusters were formed. The results are shown in Figure 6.

3.4 Source reconstruction

Figure 7 shows the values of the weight matrix W along with the values of the product of S and W (reconstructed signal). Although it is hard to determine how well the technique performed from this figure, it shows an overall picture of where the errors are. A ground truth weight matrix (W_{truth}) was created based on the labeled events, and used to evaluate the performance of the algorithm. The results of this comparison, by appliance, are shown in Table 4 where the row-sums for each of these matrices are compared.

Table 4 Presence points of pure clusters vs. ground truth appliances

Cluster/ Appliance	∑ of presence points	∑ of ground truth points
1/157 (Coffee machine)	3885	2000
5/70 (Oven – upper)	13923	24560
8/155 (Toaster)	26481	6855
11/72 (Garbage disposal)	1669	1225

We observed, by this table and Figure 7, that even the pure clusters have some offsets in terms of presence points, when compared to their ground truth. Overall, the obtained signal is not far from the original one, but this was obtained at the cost of some unrealistic behavior of the sources detected.
The most faithful tracking of an appliance is, by far, obtained in the case of appliances 70 (Oven – upper) and 72 (Garbage disposal). The results for the other appliances remain at rather modest levels.

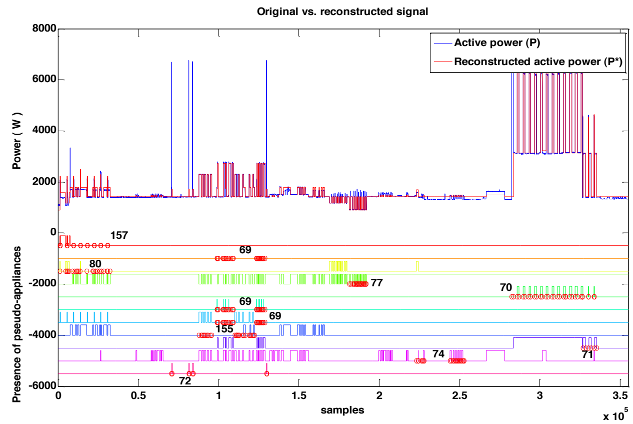

Figure 7 Original vs. reconstructed signal

4. DISCUSSION

This work is divided in three parts: detection of events, detection of appliances (clustering and coincidence detector fall under this same category) and detection of the states of appliances. The most relevant discussion however is on the clustering and reconstruction.

4.1 Clustering

There were two main issues with the clustering algorithm: different appliances were sometimes clustered together and some appliances were broken down between several clusters.
The first is mainly due to similarities in terms of consumption levels. This is the case for cluster 3, which included various appliances 80 (Kitchen Under Cabinet Lights), 81 (Range Hood Fan), 86 (Kitchen Hallway Overhead Lights), 158 (Refrigerator Light) and 182 (Basement Stairs Light); and also cluster 10, which included appliances 74 (Breakfast Area Under Cabinet Lights), 75 (Breakfast Area In Cabinet Lights), 76 (Breakfast Area Hanging Lights), 78 (Kitchen Hanging Lights) and 84 (Range Hood Light). It is worth noticing that many of these appliances are lights. This is a relevant limitation of our approach as we are unable to disaggregate between appliances that are this similar. The underlying reason may be the choice of features.
The second issue has to do with multi-state appliances. This is mainly the case with appliance 69 (Microwave) and 155 (Toaster), which show multiple states and generate several clusters. This indicates that the FSM (finite state model) of an appliance [9] are being reduced to a summation of two-state appliances, as also pointed out by [10].
We make one last observation: the clustering returns slightly different results every time, due to the random nature of the chromosomal mutations. These variations, however, did not cause major differences in the final results.

4.2 Matching Pursuit

The main sources of errors for the matching pursuit algorithm come from:

i) false negatives by the event detector, and

ii) the limited ability to explain an event with the clusters that were found.

Regarding the event detector, if it fails, the error will only endure until the next event is detected. If the rate at which events are occurring is low, this may have a big impact in terms of the prediction of energy consumption as the error can persist for longer.

Another problem we found is that because every step of the matching pursuit algorithm is performed without taking into account the results of the previous one, some of the rows of the weight matrix W contained periods of rapid switching between ON and OFF. We suspect that additional constraints to increase the sparsity of the resulting matrix, and prevent this situation, may increase the accuracy of the algorithm.

It is not obvious to us that this approach provides the best solution when applied to the chosen features (P and Q). We noted that for 'isolated' clusters (on the PQ plane) the results on the reconstruction were encouraging (see appliances 70 and 72 in Figure 7, as they are clearly separated from everything else, as shown in Figure 4). Therefore, our initial judgment of these features' viability remains rather reserved. Nevertheless, we are motivated to experiment with other traits, such as spectral coefficients deltas [11], for example.

5. CONCLUSION

We proposed an unsupervised approach to appliance disaggregation that contrasts with existing supervised approaches by not requiring prior knowledge on the set of present appliances. To validate our preliminary results we used a dataset that presented ground truth for 18 appliances over a period of 1.5 hours. The results indicate that a clustering solution close to the real appliances is possible primarily for big appliances.

This does not happen for smaller appliances (or very similar in terms of consumption) as they will be join together.

On the other hand, multistate appliances can be easily modeled as a set of two-states appliances. Also, the matching of the ground truth is mainly achieved in the case of a few of the big appliances.

We were mainly interested in the feasibility of such an approach as well as its performance with the chosen features (real and reactive power). We remain optimistic about the developed tools but conclude that the features mentioned above may not be the best in achieving the desired separation between classes, and plan to investigate others in our future work.

6. ACKNOWLEDGEMENTS

The authors would like to gratefully acknowledge the support from the Robert Bosch LLC Research and Technology Center North America and the National Science Foundation (NSF) grant #09-30868. The opinions expressed herein are those of the authors and not of the NSF.

7. REFERENCES

[1] Bergés, Mario , Ethan Goldman, H. Scott Matthews and Lúcio Soibelman(2008) Training Load Monitoring Algorithms on Highly Sub-Metered Home Electricity Consumption Data, Tsinghua Science & Technology, 13(1): pp 406-411

[2] Bergés, Mario, Ethan Goldman, H. Scott Matthews, and Lúcio Soibelman. "Learning Systems for Electric Consumption of Buildings". ASCE International Workshop on Computing in Civil Engineering, Austin, Texas, 2009.

[3] C. H. Chou, M. C. Su, and E. Lai, "A new cluster validity measure and its application to image compression," Pattern Anal. Appl., vol. 7, no. 2, pp. 205–220, Jul. 2004.

[4] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze, "Hierarchical clustering," in An Introduction to Information Retrieval, n.d., http://nlp.stanford.edu/IR-book/pdf/17hier.pdf.

[5] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze, "Single-link and complete-link clustering," in An Introduction to Information Retrieval, n.d., http://nlp.stanford.edu/IR-book/pdf/17hier.pdf.

[6] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze, "Evaluation of clustering," in An Introduction to Information Retrieval, n.d., http://nlp.stanford.edu/IR-book/pdf/17hier.pdf.

[7] Das, S. & Abraham, A. & Konar, A. (2008) Automatic Clustering Using an Improved Differential Evolution Algorithm. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 38 (1) : 218-237

[8] Dong Luo et al., "Monitoring HVAC Equipment Electrical Loads from a Centralized Location - Methods and Field Test Results," ASHRAE Transactions 108, no. 1 (2002): 841-857.

[9] Hart, G., (1992). Nonintrusive appliance load monitoring. Proceedings of the IEEE, 80(12), p.1870-1891.

[10] Hyungsul Kim, Manish Marwah, Martin Arlitt, Geoff Lyon and Jiawei Han, "Unsupervised Disaggregation of Low Frequency Power Measurements", SIAM International Conference on Data Mining (SDM 11), Mesa, Arizona, April 28-30, 2011.

[11] Jin, Yuanwei, Eniye Tebekaemi, Mario Bergés, Lúcio Soibelman. "A Time-Frequency Approach for Event Detection in Non-Intrusive Load Monitoring". Signal Processing, Sensor Fusion, and Target Recognition XX, Orlando, Florida, 2011

[12] Rowe, Anthony, Mario Bergés, Raj Rajkumar. "Contactless Sensing of Appliance State Transitions Through Variations in Electromagnetic Fields". 2nd ACM Workshop On Embedded Sensing Systems For Energy-Efficiency In Buildings (BuildSys 2010), Zurich, Switzerland, 2010.

[13] S. Mallat and Z. Zhang (1993) Matching pursuit with time-frequency dictionaries. IEEE Transactions on Signal Processing, 41:3397-3415.

[14] Te-Won Lee, Michael S. Lewicki, Mark Girolami, and Terrence J. Sejnowski, "Blind Source Separation of More Sources Than Mixtures Using Overcomplete Representations" (n.d.), cnl.salk.edu/~tewon/Public/ocica99.pdf