Appliance-specific power usage classification and disaggregation

Srinikaeth Thirugnana Sambandam, Jason Hu, EJ Baik
Department of Energy Resources Engineering Department, Stanford Univesrity
367 Panama St, Stanford, CA 94305

I. Introduction

Energy disaggregation (also referred to as non-intrusive load monitoring) is the task of inferring individual loads of the system from an aggregated signal. Appliance-specific energy usage feedback provides consumers with a better understanding of the impact of their consumption behavior and may lead to behavioral changes that improve energy efficiency. Studies have shown consumers improved efficiency by as much as 15% after getting direct feedback of this type [1]. Once a signal is disaggregated, the signals need to be classified according to the appropriate appliance. With the increasing interest in energy efficiency and recent relevance of machine learning, there is a lot of potential for both predicting and classifying appliance-specific load signals using a wide range of machine learning algorithms. In this study, we utilize a publically available dataset of power signals from multiple households to disaggregate, then classify appliance-specific energy loads.

There have been several previous works that have discussed disaggregating energy signals including Kolter and Johnson (2011) [2], Faustine et al. (2017) [3], Kelly and Knottenbelt (2015) [4], Fiol (2016) [5], as well as a comprehensive study conducted by the Pacific Northwest National Lab on Characteristics and Performance of Existing Load Disaggregation Technologies [6]. Based on these literatures, we have deduced that the two main methods that are used for energy signal disaggregation are Hidden Markov Models, as used in Kolter et al. (2011), and Deep Learning methods such as Artificial Neural Networks, as used in Kelly and Knottenbelt (2015).

Artificial Neural Network (ANN) is an effective method which automatically learns and extracts a hierarchy of features from the signals and disaggregates them according to the distinct features of an appliance. What is unique about ANN is that once the data is learned, the computer does not need ground truth appliance data from each house to disaggregate the energy signals. However, the training process is computationally heavy. On the other hand, a Hidden Markov Model (HMM) is a Markov Model with each state characterized by a probability density function describing the observations corresponding to that state. In a Hidden Markov model there are observed variables and hidden variables. Given the limitations in our computational power, we focus our study on building an HMM model for energy disaggregation.
In our project where we intend to apply Hidden Markov Models to disaggregate total electricity consumption data to the individual appliance level.

Once we have disaggregated the signal, we need to classify the different separated signals to the appropriate appliance. There have also been several studies classifying appliance-specific energy loads. Mocanu et al. (2016) [7] compares four different classification methods, including Naïve Bayes, k-Nearest Neighbors (KNN), and Support Vector Machines (SVM). Another study Altrabalsi et al. (2015) [8], combines k-means with Support Vector Machines to also classify energy signals in a simplistic manner. Similarly, in our study, we aim to apply these aforementioned supervised learning techniques to individual appliance level data and compare the results of multiple classification methods.

II. Data and Data Processing

The dataset used for this project is the Reference Energy Disaggregation Dataset (REDD). This dataset contains total electricity consumption data from 6 households, and appliance-specific consumption data from 268 appliance loads within those households, over a total of 119 days[5]. The data is sampled every three seconds, resulting in a sizeable data set. Below is a visualization of the power usage of variance appliances from one household throughout one day. Note the high intermittency of the data as well as seemingly random spikes of energy usage from different appliances.

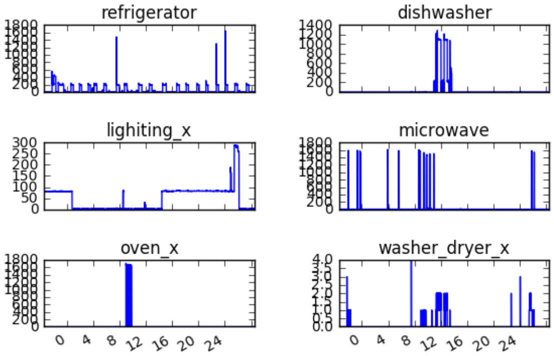


Figure 1. Visualization of load from one household over the course of a day (Vertical axis indicates power [W])

From this dataset, we assume that the most useful features for each appliance, is the maximum and minimum power value of the day, the mean and variance of the power of the appliance over the course of an hour, the baseline value of the appliance by day, and the weekday, hour, minute, and second that the appliance is operating. The baseline value of the appliance was extracted using the tools in the *peakutils* package in Python.

III. Methods

A. Classification Methods

For classification, we compare multiple methods over various scenarios of data sets. We use two household data (House 1 and House 2) for classification. The three scenarios explored are: Type 1. Train and test on House 1 data; Type 2: Train and test on aggregated House 1 and House 3 data; Type 3: Train on House 1 and test on House 3 data. The following design allows us to understand not only the effectiveness of different methods, but also the effect that the data might have on those results. We are particularly interested in whether the classification methods would be able to perform well given individuality of household appliance usages. We only consider four different appliances that had four distinct patterns, including refrigerator, bathroom outlets, and two different lights.

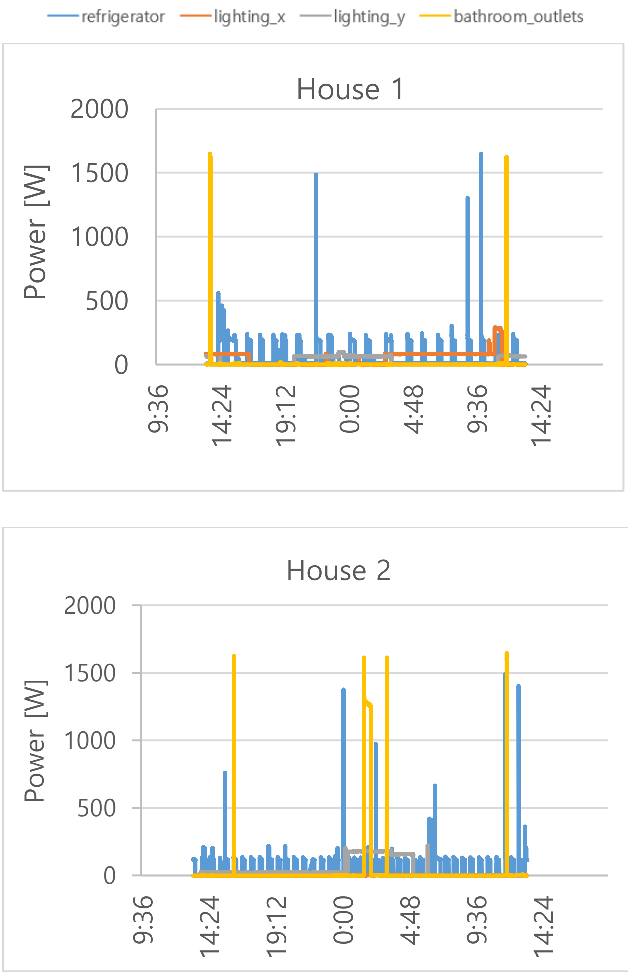


Figure 2. House 1 and House 2 appliance loads over one day

The different classification methods we compare are Naïve Bayes, Support Vector Machines, and K-nearest neighbors, all of which we learned in class. The models were implemented using the *sklearn* package in Python.

B. Disaggregation Methods

Hidden Markov Models (HMM) were used for the purpose of disaggregation. HMM is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (i.e. hidden) states. The hidden Markov model can be represented as the simplest dynamic Bayesian network as shown in Fig.3. Here, $g^0(t)$, $g^1(t)$, $g^2(t)$ are the hidden and unobserved states and $x(t)$ is the observed state at time t. Note that in this figure, there are three hidden states and one observation state. In a more general formulation, there can be multiple such states. A HMM is parametrized by:

- State transition probabilities **(A)**: The matrix where the $(i,j)^{th}$ entry is the probability of transitioning from hidden state i to j
- Emission probabilities **(B)**: The nature of the probabilities of the hidden states given an observed state

Each hidden state in the model is represented by a probabilistic function, and in our project we modelled it as a mixture of Gaussian distributions. To train the parameters of the HMM model, we solve the following:

$$A, B \;=\; \arg\max_{A,B} \sum_{\vec{z}} Q(\vec{z}) \log \frac{P(\vec{x}, \vec{z}; A, B)}{Q(\vec{z})}$$

This is solved by applying the EM algorithm which we learnt in class. ($\vec{z}$ is the hidden state)

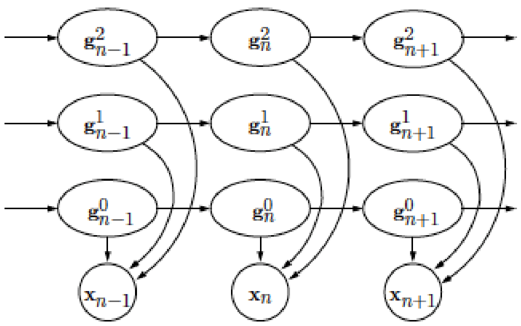


Figure 3. Visualization of HMM

Two separate HMMs were constructed for the purpose of disaggregation – the individual appliance model and an aggregated model.

In the individual model each appliance is modelled separately with a specific number of hidden states (based on an examination of the power consumption levels of the device). The HMM learns the transition probability matrix (A), and the mean and variance of the Gaussian distributions of each hidden state based on the power signal of each specific device (training).

The aggregated model is used to correlate the behavior of each appliance to the aggregated load. This is accomplished by the following HMM formulation:

- States: The possible states are cross-product of states of each individual appliance. For instance, if appliance 1 has states S_1, S_2, S_3 in its individual model and appliance 2 has states T_1, T_2, then the combined HMM will have states: $\{(S_1, T_1), (S_1, T_2), (S_2, T_1), (S_2, T_2), (S_3, T_1), (S_3, T_2)\}$
- Input/observations: The input is a single variable that takes value of the total energy consumption load.

The motivation behind using this architecture is as follows. First, we train individual appliance models assuming they are independent. This helps us understand how many states each appliance needs to be modeled accurately because individual appliances are trained using individual loads and not the aggregated load. Then we learn co-relations between appliance loads through this combined HMM which ties the underlying states of the individual appliances (modeled after individual loads) to the total aggregated load.

The aggregated model is initialized by combining the learnt parameters from the multiple individual appliance models. This was done using Kronecker multiplication, which is common in graph generation, but directly applies to our problem.

IV. Results and Discussion

A. *Classification*

The table below summarizes the results for different classification methods on the different scenarios explored.

Table 1. Accuracy of classification methods

		TYPE 1	TYPE 2	TYPE 3
NB	Training Set	0.624	0.445	0.624
	Test Set	0.626	0.444	0.143
SVM	Training Set	0.810	0.566	0.810
	Test Set	0.811	0.565	0.211
KNN	Training Set	0.997	0.981	0.997
	Test Set	0.992	0.960	0.153

Overall, KNN shows the most accurate over all the different classification methods used for all different scenarios (which is consistent with the results from Mocanu et al. (2016)). This is most likely because KNN captures the non-linearity of the power load, while SVM is limited to linear classification. An appliance power load is a mixed integer problem that does not have apparent linear tendencies. Naïve Bayes was the worst classifier, which was expected given that the power usage of different appliances within one household are not completely independent from each other.

The significant drop in accuracy from Type 1, to Type 2, to Type 3 over all the classification algorithms highlights the strong individuality that exists in different household appliance usage data. This makes sense, as in we would expect a household one young person to a large family to have different energy usage profiles throughout the day. This has large implications for the task of applying energy classification, disaggregation, and regression to a much wider audience in the context of demand response or load management from the utilities.

B. Disaggregation (contribution % estimate)

For disaggregation, the train and test data were only from House 1. The train test-split used was 80-20%. We used the Python library *hmmlearn* to build the HMMs that were described.

One metric for defining the effectiveness of disaggregation is to compute the % of power predicted by the algorithm for a given aggregated load and compare it with the actual contribution from the appliance. The summary of results in that format is presented in Table 2. Two different HMM based algorithms were attempted for the disaggregation. The first one consisted only of building the individual appliance models and applying them on the aggregated load profile to determine the contribution of each appliance. This however, led to the algorithm over-predicting the contributions from each appliance (Tab.2 (a)).

The idea to use a second aggregated HMM model (as described in the previous section) was formed to handle these over-estimates and the results for this updated model is also presented (Tab.2 (b)).

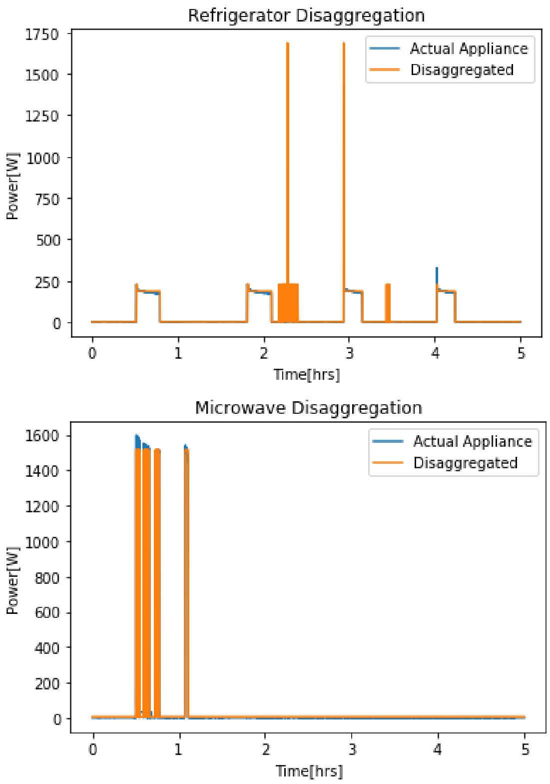
Table 2. Summary of the results of the two disaggregation method

(a) Appliance	**Actual energy (%)**	**Estimated energy (%)**
Refrigerator	77.95	91.89
Microwave	22.05	25.47

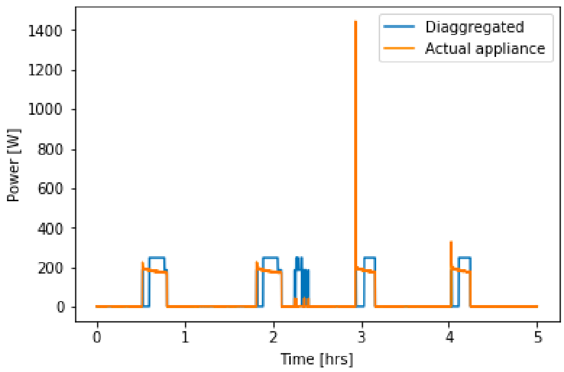
(b) Appliance	**Actual energy (%)**	**Estimated energy (%)**
Refrigerator	77.95	78.37
Microwave	22.05	16.57

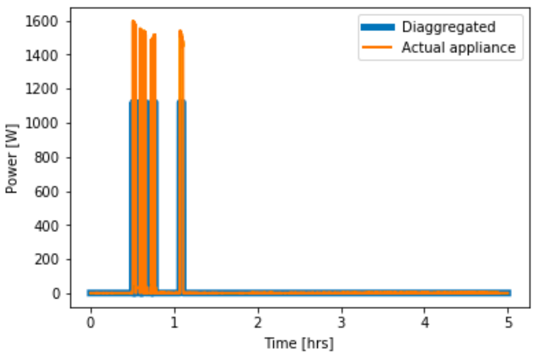
C. Estimate of appliance signal

After the HMM has estimated the most likely combination of states for a given input, it is possible to obtain an estimate of what the algorithm expects the disaggregated signal would look like by sampling the probability distribution of the corresponding hidden state at each time step. This ‘predicted’ signal (from the test set) can be compared to the actual appliance signal in that duration and this is presented in Fig. 4 for the two algorithms.



(a)





(b)

Figure 4. Comparison of appliance signals results for refrigerator and microwave using the simple HMM (a) and the updated algorithm (b)

D. Qualitative analysis: Disaggregation

From the disaggregation results shown in Tab.2, we can see that the simple model tends to over predict when it is tested on aggregated load signal. This is because the individual appliance models were trained *only* on the separate appliance curves. The model had never encountered any form of aggregated load and thus doesn't perform well. This is virtually like overfitting to the appliance data. The aggregated model however, overcomes this by defining the hidden states as combinations of the individual hidden states.

From Fig.4, it seems that the simple model captures the *periodic* spikes better than the updated model. This is true, primarily because the simple model assumes the refrigerator has more hidden states than the updated model (4 vs 3). However, this is also why the simple model tends to predict spikes in places they're not actually present. The number of hidden states chosen for the updated model ensures that it captures the necessary features without over-fitting.

V. Conclusions

To summarize our report, we conducted a general study on household power usage data to first classify and identify different appliances by their unique signals, then further performed a disaggregation to extract those individual appliance-specific load usages.

KNN proved to be the most effective classification algorithm that captured all the non-linearities of the data. However, as a result of significant variations in household consumption profiles, classification algorithms trained on multiple houses or trained by one house and tested on other houses perform poorly.

Disaggregation using a two-step modeling approach outperforms simple HMMs trained on individual appliances. Predicted signal behavior was also compared between the two methods.

VI. Challenges & Future Work

For next steps in classification, we would be interested in doing a more thorough feature extraction to capture more of the appliance-specific load behaviors. We would also be interested in extending the study to all the households in the dataset and comparing its performance relative to training on only two households. Such work would help us establish whether the variation between households is extreme enough that it is extremely unlikely to be capture or there is some commonality between the household appliance usages.

During disaggregation, one challenge with building the aggregated HMM was that the number of states increase exponentially as the number of appliances increase, and was computationally infeasible after a point. Efficient representation of these states would help disaggregate larger combinations of appliances.

A further refinement of the aggregated HMM would be to define custom emission probabilities to ensure that the disaggregated loads are not wrongly estimated. One way to do that would be to add a constraint that the appliance load at each time step cannot be higher than the aggregated load.

Contributions

All three authors came up with the idea of working with energy disaggregation and classification. All three authors also worked together in preprocessing the dataset.

EJ Baik worked on implementing the different classification methods, while Jason and Srinikaeth worked on designing and implementing the disaggregation method.

Jason and Srinikaeth worked heavily on the poster and presentation (EJ Baik was absent due to her presentation at a conference) while EJ Baik focused on formatting and writing up the final report.

Overall, all the teammates were happy with each other's contribution to the project and confident that they made a strong team together.

References

[1] B. Neenan, "Residential Electricity Use Feedback: A Research Synthesis and Economic Framework," *Manager*, pp. 1–8, 2009.

[2] J. Z. Kolter and M. J. Johnson, "REDD : A Public Data Set for Energy Disaggregation Research," *SustKDD Work.*, vol. xxxxx, no. 1, pp. 1–6, 2011.

[3] A. Faustine, N. H. Mvungi, S. Kaijage, and K. Michael, "A Survey on Non-Intrusive Load Monitoring Methodies and Techniques for Energy Disaggregation Problem," 2017.

[4] J. Kelly and W. Knottenbelt, "Neural NILM: Deep Neural Networks Applied to Energy Disaggregation," 2015.

[5] A. Fiol and C. J. Castro, "Algorithms for Energy Disaggregation Director : Josep Carmona," 2016.

[6] E. Mayhorn, G. Sullivan, R. Butner, H. Hao, and M. Baechler, "Characteristics and Performance of Existing Load Disaggregation Technologies," no. April, 2015.

[7] E. Mocanu, P. H. Nguyen, and M. Gibescu, "Energy disaggregation for real-time building flexibility detection," *IEEE Power Energy Soc. Gen. Meet.*, vol. 2016–Novem, 2016.

[8] H. Altrabalsi, L. Stankovic, J. Liao, and V. Stankovic, "A low-complexity energy disaggregation method: Performance and robustness," *IEEE Symp. Comput. Intell. Appl. Smart Grid, CIASG*, vol. 2015–Janua, no. January, 2015.

Python Tools Used:

From sklearn package

linear_model.LogisticRegression()

naive_bayes.BernoulliNB()

neighbors.KNeighborsClassifier()

test_train_split()

From hmmlearn package

hmm

From matplotlib package

cm()

pyplo()t

dates() - YearLocator, MonthLocator

Used pandas, numpy, and math