Residential Electricity Disaggregation: Identifying Appliance Usage Patterns

David Newcomb
Management Science & Engineering
Stanford University

Introduction

The complex modern day issues of energy and sustainability can be approached from several angles. Addressing such large scale issues requires not only large scale fixes, but also small scale solutions. One particularly interesting and promising area of small scale research involves residential energy use. In the United States, residential energy consumption represents 21% of total primary consumption, so reducing household energy use can make a big impact on a significant slice of consumption.

One current obstacle in the way of energy reduction is that the average homeowner only sees the total electricity usage each month, and has no clear idea about which appliances are using how much energy or when. This lack of information makes it somewhat difficult for the homeowner to implement or quantify concrete energy-saving measures. Fortunately, knowledge of appliance-specific energy use has proven to effectively increase user energy efficiency by up to 15%. *The primary point of interest in this project, therefore, is to use machine learning techniques on the whole home power signal to provide the average homeowner with accurate, appliance-specific breakdowns of energy usage.*

Since it is economically infeasible to monitor literally every appliance in every home, it is more reasonable to work with data that could be provided by a basic smart meter. Machine learning techniques can then be applied to low frequency measurements of the whole home power signal in an effort to disaggregate the total electricity usage profile into its components. This process, called Non-Intrusive Load Monitoring (NILM), hopes to make the most out of a relatively sparse amount of residential energy data [1].

Research & Related Work

A significant part of the NILM research put forth thus far has involved unsupervised techniques like Hidden Markov Models, k-means clustering and Support Vector Machines (SVM). Hidden Markov models can help successfully identify the presence of low-power appliances in the presence of high loads, whereas k-means and SVM classifiers have been better at helping identify the more prominent profiles [2]. Several approaches have been suggested and tested using supervised frameworks as well. For instance, the Finite State Machine has been successfully applied to the whole home power signal by building a model for each appliance based on active and reactive power. Then, these learned appliance profiles are applied to new data and one can infer the whether a given appliance profile is present.

Ideally, one would want to create the most accurate model while requiring the least amount of information - this is likely to be an efficient choice. An unsupervised method working with unlabeled data that can perform with extremely high accuracy is the main aim for a scalable and cost-effective solution. Without the need for prior data or prior knowledge, such methods would be well suited for most homes. On the other hand, supervised learning techniques necessitate some prior dataset or some labeling of the energy usage of specific home appliances in order to function properly. Currently, the number of datasets with such relevant data is small, but research on the subject is growing rapidly, along with more publicly accessible data. Another upside is that typically the supervised approaches perform with higher accuracy.

Most research, however, has not yet gone toward a combination of effective supervised and unsupervised approaches. Often, combining a pre-processing step of unsupervised learning with a supervised analysis step proves to be very effective, and such a method could help increase disaggregation accuracy. My paper will explore the benefits and drawbacks of a pre-processing step of k-means clustering in conjunction with Hidden Markov Models that are built in a supervised manner.

Dataset Background

Currently, there is one dataset aimed at opening up the question of disaggregation, called the Reference Energy Disaggregation Dataset (REDD). It is an open dataset created by MIT professors and it tracks the energy usage of real homes by delivering a signal representing whole home energy use and up to 24 separate subsignals representing various home appliances. The data, therefore, have both training and testing components [3].

The information provided by REDD gives access to many key components of residential electricity usage. The dataset contains several weeks of home electricity use data for six houses in the greater Boston area with an extreme level of granularity: the given measurements of voltage, current and power occur hundreds of times per second. Not only does the dataset include low frequency signals for power used by the whole home and power used by 24 measured individual appliances, but it also provides high frequency AC waveform data for the whole home usage.

These high frequency instantaneous voltage and current waveforms, measured at a rate of 275 samples per second, contain enough information to extract the power load signatures of the home at any given point. Identifying these power load signatures, in addition to finding the points during which an appliance is turned on or off, allows for an initial determination of the load signatures of individual appliances. Therefore, an important component of disaggregation analysis is Switch Event Detection. Determining the timing of when an appliance changes state from on to off will start to determine the structure of the data as well as which waveform profiles correspond to which appliances. I began my analysis of the dataset, however, with an exploration of appliance profiles and used an unsupervised learning approach to approximate the profiles of each appliance.

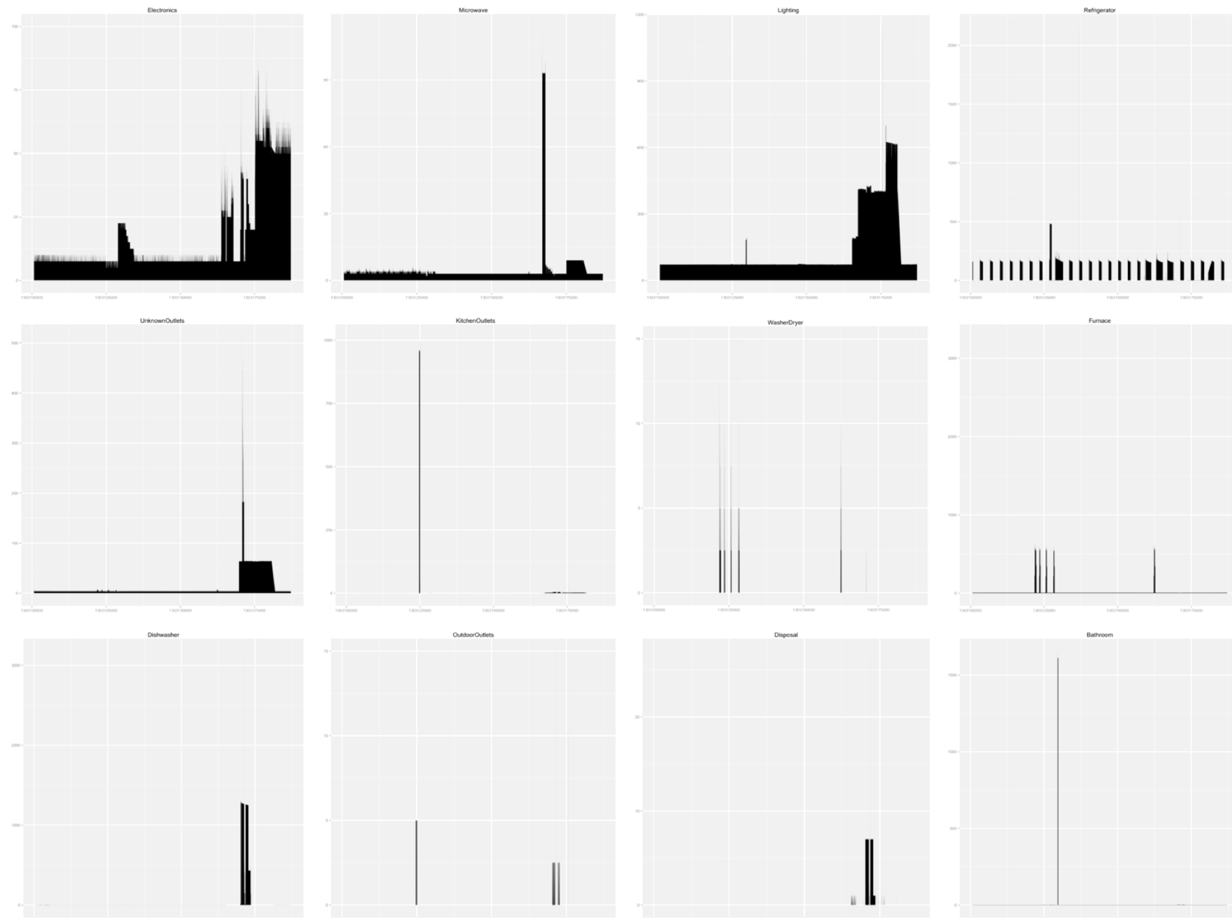

Figure 1: 24 Hour Profiles: **Row 1** (Electronics, Microwave, Lighting, Refrigerator) **Row 2** (Unknown Outlets, Kitchen Outlets, Washer/Dryer, Furnace) **Row 3** (Dishwasher, Outdoor Outlets, Disposal, Bathroom)

Load Signature ID

The graphs in Figure 1 indicate the actual load profiles of various appliances throughout a 24 hour time span, as measured in Home 5 of the REDD data set. I chose this home to work with primarily because it contained the greatest amount of information. Not only did it have both high frequency and low frequency measurements for the home, but it also provided 24 separate appliance power measurements. As seen in the profiles, there is a huge variety of profile types, ranging from the predictably intermittent refrigerator to the sparsely spiked washer/dryer. Having this data gives a clear benchmark for the unsupervised clustering analysis of appliance profiles.

Now, in order to determine the load signatures or profiles of individual appliances, it is necessary to calculate and understand how much power an appliance might be using. Since REDD provides AC waveforms for current and voltage for the whole home power signal, in addition to the real power (average power) used by the home, I could calculate two important and distinct components of power in an AC situation: reactive power and apparent power. In the case of an AC current with no reactance, multiplying instantaneous voltage and current would yield real power. In this situation there would be no excess power. However, in our real world home situation, there is reactance, so there will be excess power alternately stored and released by inductors and/or capacitors in the system. This excess power is known as reactive power and can be calculated in the following manner, where S is apparent power and P is real power:

$$N = (S^2 - P^2)^{1/2}$$

Apparent power, in this situation, is the instantaneous current multiplied by instantaneous voltage. For the data of the homes of REDD, I calculated apparent power, and used the real power given by the low frequency data to find reactive power. Given these values, I performed k-means clustering on both the absolute whole home power usage and the change in whole home power usage. The impetus behind the clustering analysis on change in power usage was the potential of being able to identify the component reactive and real power mixes for individual appliances. Then I could check whether these power mixes could be used to consistently determine which appliances were being turned on and off in the home.

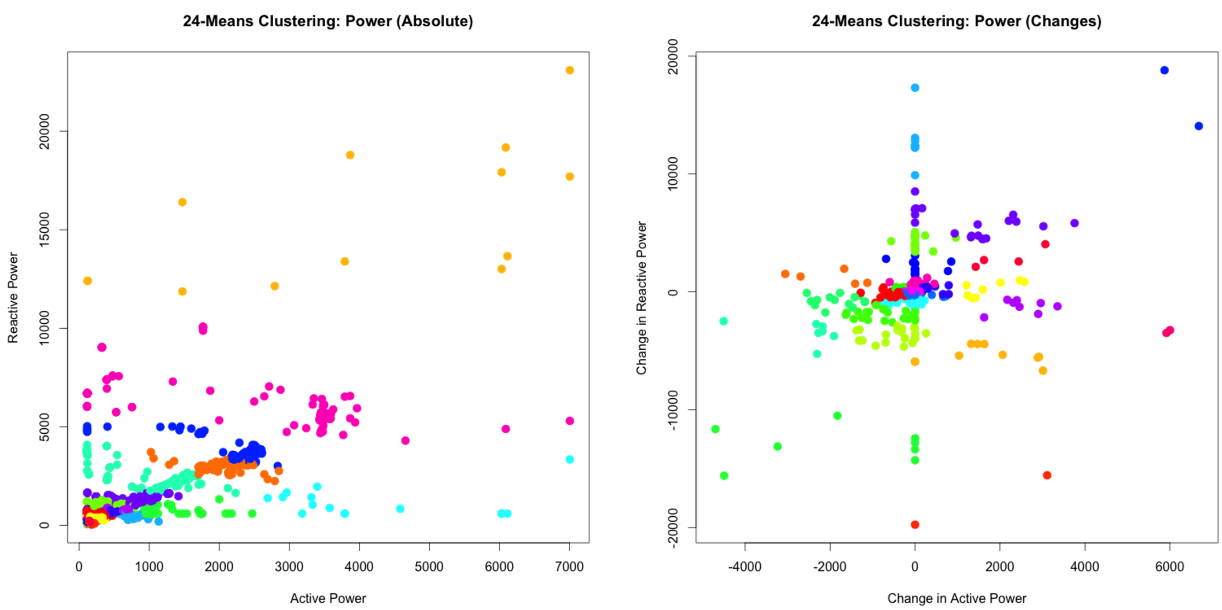

Figure 2: Comparing k-means Clustering on Absolute Power vs. Change-in-Power Data

I chose $k = 24$ due to prior knowledge that there are 24 appliances being monitored in home 5 of REDD, an important supervised edit to an otherwise unsupervised process that will be explored in detail. Figure 2 above shows the graphical results of 24-mean clustering analysis for both absolute and change in whole home power usage. While some patterns emerge distinctly for the absolute power clustering, the change-in-power clustering remains clearly spread out. I scaled each within cluster sum of squares by the size of the cluster then divided by the geometric mean of the means of reactive and active power to find comparable metrics of the goodness-of-fit for the clustering. The results corroborated the eye test showing that, by this metric, the relative cluster goodness-of-fit for absolute power was 511 times more cohesive than the loose clustering of change-in-power data (1992 vs. 1,019,419). Due to this enormous discrepancy, I decided to move forward with the clusters produced from the absolute power clustering analysis.

After determining that k-means clustering would be more effective on absolute power data, I performed this goodness-of-fit cluster analysis for several values of k and found that a few significantly lower numbers of clusters outperformed the fit from $k = 24$. The results of this analysis are shown in Figure 3 below. Whereas $k = 14$ onward matches the expected trend of an increase in fit (i.e. decrease in error) coming from $k = 5$, the group of values wherein $k = \{11, 12, 13\}$ shows a severe drop in error that breaks the trend.

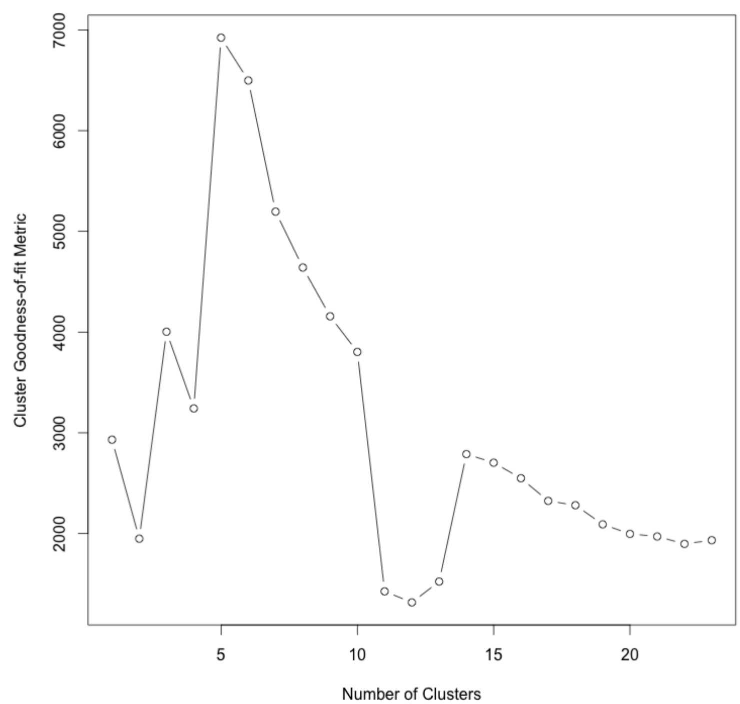

Figure 3: The goodness-of-fit metric for varying k

After looking at the data collected, it becomes clear that there are several measurements of lighting and unknown outlets which have similar profiles and ought to be part of the same cluster. Therefore, in my continued analysis, I use $k = 12$ as a better, more accurate number of clusters to use. After clustering the observations in this manner, the next necessary step was to relate the newfound cluster-based appliance profiles to specific moments when an appliance switched on or off. This Switch Event Detection was performed on the low frequency whole home signal using a supervised Hidden Markov Model.

Switch Event Detection

Hidden Markov Models (HMM) belong to a class of models called dependent mixture models. The fundamental assumption of a dependent mixture model is that at any point in time, each observation is distributed as a mixture with n components (or states). Further, time-dependencies between the observations are due to time-dependencies between the mixture components (i.e., transition probabilities between the states) and these latter dependencies are assumed to follow a first-order Markov process. An HMM, therefore, provides a specific kind of probability distribution over sequences of vectors.

In order to estimate both the parameters of this distribution and the transition probabilities, we use the expectation-maximization (EM) algorithm, which iteratively maximizes the expected joint log-likelihood of the parameters given the observations and states. Each step of the algorithm provides new parameters which we use to find a new log-likelihood and we repeat until the log-likelihood converges within a certain tolerance. The downside of the EM algorithm, and by extension the HMM, is that it runs in exponential time and is very computationally expensive. Therefore, it is often necessary to pare down the number of states or do some pre-processing of the data when trying to apply an HMM to a large amount of data.

Once we have the final parameters and transition probabilities, the last step of this supervised HMM process is to determine the most likely state associated with a certain input vector. Given the HMM distribution, running the Viterbi algorithm on the sequence of data extracts the most likely state of the system at every input [5]. This process culminates in a clustering of the input sequence into n classes and gives distinct points in time that relate to switching an appliance on or off. Figure 4 gives a graphical explanation of the resulting hidden state variables (S_t) and associated time series observations (Y_{ti}) that this supervised HMM process produces.

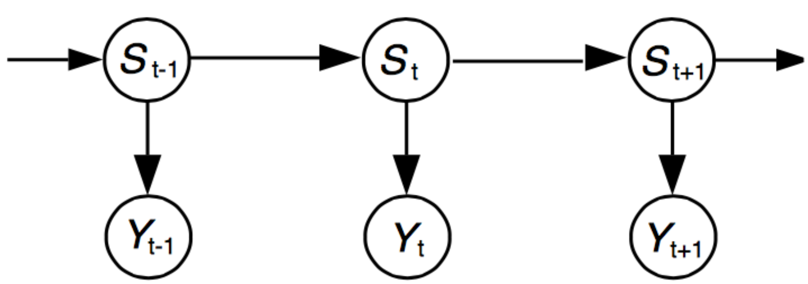

Figure 4: Hidden Markov Model Schematic

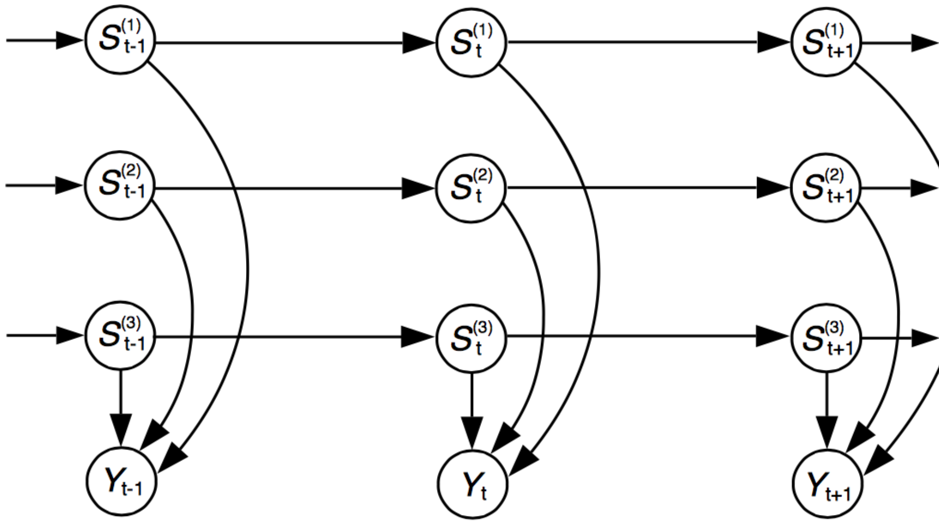

Figure 5: Factorial Hidden Markov Model Schematic

As an illustrative example of an extended HMM process, if a home signal were comprised of only 2 appliances, a water heater and a refrigerator, then we would have $n = 2^2 = 4$ states. Considering 0 = Off and 1 = On, the specific state vectors would be 00, 01, 10, and 11 (for clarification: heater On, fridge Off = 10). Technically, this type of HMM is an extension into Factorial Hidden Markov Models (FHMM) - rather than having n states for n appliances, we are examining combinations of states into the aggregate signal and therefore need to exponentiate to get 2^n states, accounting for on/off states of each appliance [6]. If we had 100 whole home power observations to work with, we would then build an HMM on the observations with 4 states. From the HMM distribution, we would use the Viterbi algorithm to determine which of the 4 state vectors we are in at any given point in our observations.

REDD Factorial Hidden Markov Model

The measurements of REDD for Home 5, our home of interest, give power observations every second for several days of data on the whole home power signal. In order to make the data more computationally manageable, I reduced the data by an order of magnitude, taking every tenth observation. This left me with a dataset of approximately 30,000 observations. To build a complete FHMM given the 24 known appliance measurements, I would need to build a model with $2^{24} = 16,777,216$ states. Making an FHMM with this number of states requires not only a long time, but also more RAM than my computer can provide.

The unsupervised clustering of high frequency power data discussed earlier shows that only 12 clusters (representing aggregate appliances) can model the system well. This reduction in effective number of appliances by a factor of 2 translates into a reduction of states by a factor of 2^{12} and a total number of states equal to 4096. Unfortunately, even this reduced number states crashed my R sessions and required more RAM than I had available.

Due to these computational obstacles, I pivoted toward running significantly-reduced-state FHMM's on pre-processed data. Rather than focus on all 24 appliance signals or even 12, due to computational complexity, I decided to graph the power consumption related to the centers of each cluster found in the unsupervised section of this analysis. Figure 6 shows that the top four appliance clusters account for essentially 75% of household power consumption, so I decided to place my focus on these cluster groups [7].

By focusing on the top three to four power consuming appliance clusters (resulting in n = 8 and n = 16 states, respectively), I could build FHMM's with no computational worry. In order to examine only the top four appliance clusters from the unsupervised initialization step, I recognized that the smallest difference of real (active) power between each of these four cluster centers was about 1000W. Since any combination of appliances creating these clusters should lie above 1000W as well, I further processed the dataset by eliminating all changes in whole home signal observations lying below this threshold. At this point, I reverted to 1Hz frequency data and was able to maintain a reasonably large amount of workable data (25,000 observations).

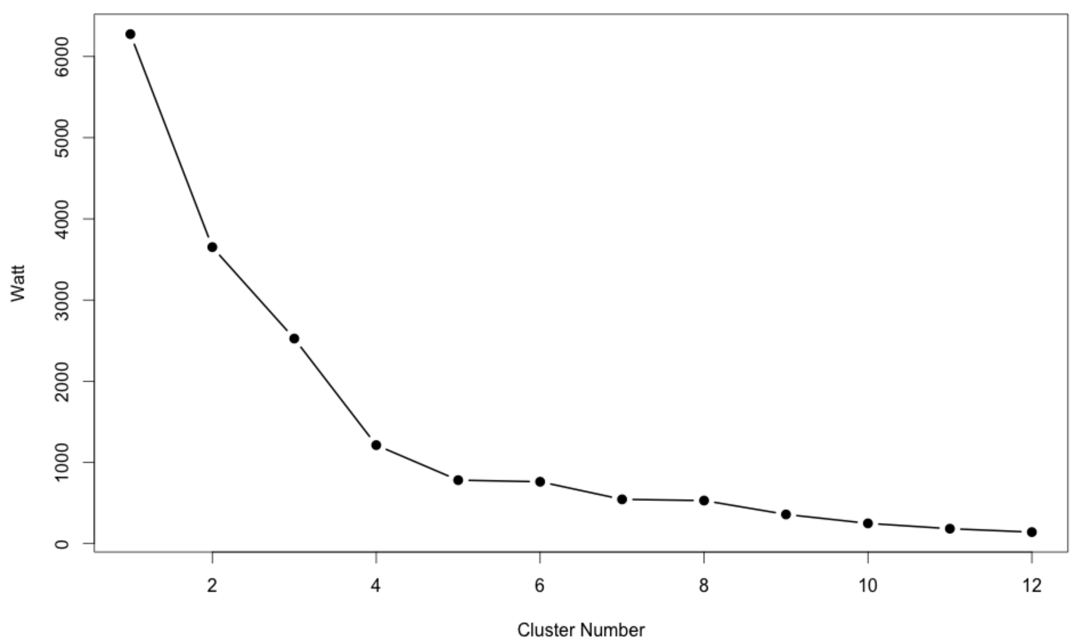

Figure 6: Ranked Power Consumption of 12-means Cluster Centers

Running the Viterbi algorithm on this final FHMM with 16 states (4 appliances) provided data on when exactly it was likely that an appliance switched from one state to the next. The probability matrix in Figure 7 demonstrates the transition probabilities between all of the states in the FHMM. Seeing as no transition probability exceeds 3%, Markov model assumption of independence between states remains intact. The only transitions close to even that small 3% maximum are transitions from states 6 to 7, 6 to 11, and 12 to 14.

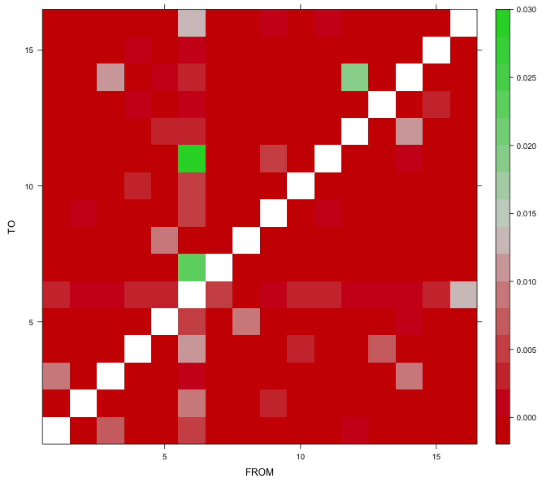

Figure 7: FHMM State Transition Probability Matrix

In order to get a better understanding of the FHMM states, Figure 8 provides a plot of the power measured at each observation, grouped by most likely associated state. Each clustered group represents a different state vector. The highest cluster represents the whole home power usage with state vector of 1111 (i.e. all top four appliances are turned on), whereas the lowest state vector of 0000 represents the situation of none of the top four appliances being turned on. As we can perhaps see in the plot, the FHMM does a decent job at the extremes of separating distinct power states of the home in relation to these top four appliances. However, there are several clusters in the midrange which approximate each other and lend confusion to the image.

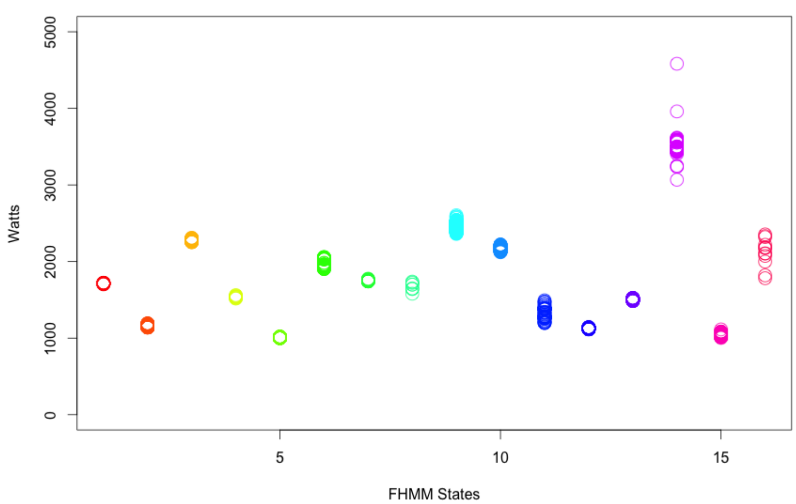

Figure 8: Power Consumption of FHMM States

Nevertheless, the FHMM does associate the whole home power signal with its most likely state vector. In order to gauge whether this FHMM is predicting anything close to the actual home situation, we first go into the individual signals for every appliance and find the top four appliances by consumption. These are the Subpanel, Furnace, Electric Heater, and Dishwasher in Home 5 of REDD. Next, we simply cross-check to see whether the times when the appliances actually turned on correspond to the times when the Viterbi algorithm applied to the FHMM expects the appliance to turn on or off. The results of this analysis are discussed further in the following section. While the FHMM fit the top two appliances decently, it did not perform particularly well when trying to fit the third and fourth top appliances.

Comparison to Actual Data

After both performing the unsupervised k-means clustering and building the FHMM, there were a few important comparisons to be made. Namely, the main points of interest were whether the unsupervised k-means clustered profiles matched the actual appliance power profiles and whether the supervised Factorial Hidden Markov Model accurately detected the presence of top four appliances in the home signal. Figure 9 shows the first comparison between actual data and clustered profiles with $k = 12$. The actual top twelve appliances represented from left to right are the (1) Subpanel, (2) Furnace, (3) Electric Heater, (4) Dishwasher, (5) Refrigerator, (6) Bathroom, (7) Lighting, (8) Kitchen Outlets, (9) Electronics, (10) Unknown Outlets, (11) Microwave, and the (12) Washer/Dryer.

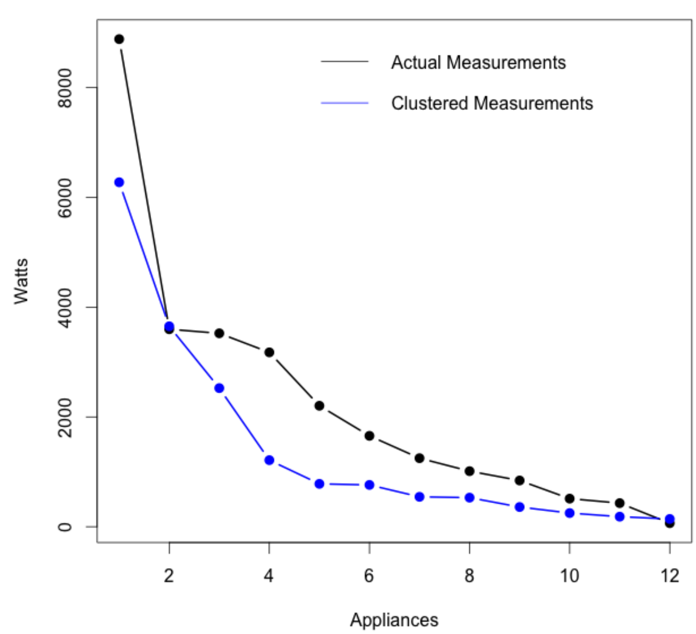

Figure 9: Comparing Cluster Power Profiles with Actual Appliance Power Profiles

While the clustering gets the general trend correct for the long tail and for the couple of appliances in the beginning, it has large errors on the Dishwasher, Refrigerator and Bathroom power loads. Looking at the data, perhaps the irregularity of a single enormous spike in Bathroom usage threw off the clustering method (my k-means algorithm is particularly sensitive to outliers since it is based on Euclidean distance). The Refrigerator load is also poorly identified likely due to its regular step function behavior: half the time its load is low (like that of Unknown Outlets) while the other half of the time it is significantly high and constant. The basic k-means algorithm may have considered these distinct parts of the same profile incorrectly as two separate appliances, thereby underestimating the power consumption of the Refrigerator. I have yet to determine causes of the low estimate for the dishwasher.

Overall, the approximation of mean power consumption by k-means clustering performed well enough to allow me to make relatively safe assumptions in the pre-processing step of my supervised FHMM. Namely, it allowed me to isolate a top four for significant appliances (based on the actual data, I would have likely expanded this to top 5 or 6), and to approximate the power usage of said appliances within a reasonable range of reality.

Regarding the FHMM performance, there were several metrics to choose from such as fraction of total energy assigned correctly, normalized error in assigned power, true/false positive rates, precision, recall, and F-score (harmonic mean of precision and recall). Since I built the FHMM on only the top 4 appliances, I found that normalized error in assigned power would be a good metric. This value is the sum of the differences between the assigned power and actual power of appliance n in each time slice t, normalized by the appliances total energy consumption. I performed this analysis for each of the top four appliances and found values of 6.42, 4.38, 19.52, and 16.91. As a benchmark, the normalized error in assigned power results from the Batra, Nipun, et. al analysis hit 1.35, which is far lower than any of my FHMM results.

Conclusion

Overall, my analysis suffered from a couple main aspects: an inability to implement my FHMM in a computationally efficient manner, and too much error in my clustering method. Both can and should be improved upon as I continue to work on this project. Figure 10 shows the whole home power signal for Home 5 of REDD disaggregated into all of its components over the course of a single day. Expanding the crux of my analysis from 4 to 24 or more appliances will allow me to disaggregate the entire signal and hopefully approximate that image from Figure 10. Further, applying the same improved analysis from Home 5 onto other homes will be a good test of whether my method has any merit.

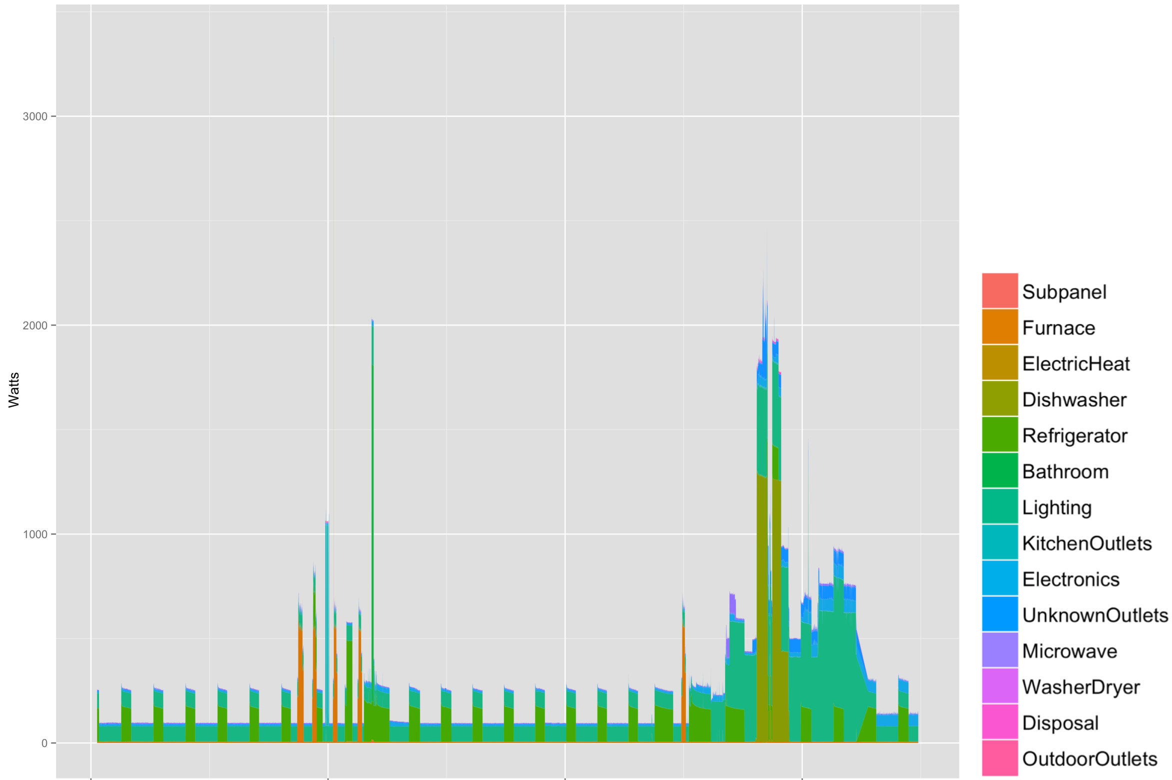

Figure 10: Components of Aggregate Home Signal (24 Hour Window)

References

[1] G. W. Hart. Nonintrusive appliance load monitoring. Proceedings of the IEEE, 80(12):18701891, 1992.

[2] Z. Ghahramani and M. I. Jordan. Hidden Markov Models. *Machine Learning*, 1997.

[3] J. Zico Kolter and Matthew J. Johnson. REDD: A public data set for energy disaggregation research. In proceedings of the SustKDD workshop on Data Mining Applications in Sustainability, 2011.

[4] A. Zoha, A. Gluhak, M. A. Imran, and S. Rajasegarar. Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey. *Sensors*, 2012.

[5] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory, 13(2):260269, 1967.

[6] Ghahramani, Zoubin, and Michael I. Jordan. "Factorial hidden Markov models." Machine learning 29.2-3 (1997): 245-273.

[7] Batra, Nipun, et al. "NILMTK: An open source toolkit for non-intrusive load monitoring." Proceedings of the 5th international conference on Future energy systems. ACM, 2014.