
Appliance Level Energy Disaggregation

Samir Sen
samirsen@stanford.edu

Fred Lu
fredlu@stanford.edu

Abstract

We build and experiment with various models for separating total electric energy usage (kWh) into usage by appliance in a given household. We formulate the task of energy disaggregation as a regression problem for each appliance, where we predict a continuous valued usage in kWh at each time step conditioned on the total energy used from m previous time steps. We note that the challenge in this task is building a predictive network that is able to learn a mapping from $\mathcal{R} \rightarrow \mathcal{R}^n$ appliances. We therefore experiment with a sparse coding model in which we attempt to learn signatures of each home appliance that together sum to aggregate energy signal. We achieve the best results on the REDD dataset (plug-wise meter data from 6 homes) using a Random Forest model with 89% accuracy on the test set as well as our sparse coding model with 91% accuracy.

Mentor: Akhila Yerukola

Code repository: <https://github.com/fl16180/disaggregate-NRG>

1 Introduction

For years, we have watched the sticker shock of utility bills affect many around us. We know there is a meter at the back of the house that measures how much electricity we consume in total as a household and we receive a bill once every month. Friends and family remain in a dilemma as to what causes these bills to be so high many of the times. We scrutinized our electric bills in detail looking for clues as to why, but kilowatt hour was aggregated for the whole month. Listening in on calls my mom made to Clark Public Utilities, she inquired about what appliances specifically could be the main cause of our high bill. Although apologetic, the call center agent could offer no help despite her best effort, besides some general energy saving tips like setting thermostats to 68 F, use efficient lighting and so on. Too general.

For our project, we experiment with and build several disaggregation models to separate total home/building electricity usage by appliance. We use the REDD dataset and formulate the task as a regression problem predicting the energy usage time series of each of n appliances given an input time series of total building energy consumption. By learning the signature for individual appliances, we hope to build a generalized model for predicting usage percentage per appliance and give an accurate estimate of a given home's energy profile.

2 Related Work

The development of machine learning approaches to energy disaggregation is a relatively new field of research. To a large extent, this is due to a scarcity of large-scale public datasets with buildings that have been submetered. For this reason, research in this area is often highly centered around specific datasets. In the past several years, some small- to medium-sized datasets have been presented, including the REDD dataset used in our analysis. Because the REDD dataset, released in 2011, is one of the oldest and most well-known, it serves as a reference standard for algorithm development and testing [1]. For this reason, we will use this dataset.

Solutions to energy disaggregation are clustered around two approaches. The first and original method is combinatorial optimization, which basically solves a subset sum problem with appliances adding up to the aggregate signal at each time interval [2]. The original method models time steps as independent, which leads to a simple optimization. Nevertheless, this is an exponential-time problem and quickly becomes intractable as the number of appliances increases [3].

A more recent model that has Factorial hidden Markov models (FHHM), an extension of the hidden Markov model for multiple states and multiple appliances [1, 4, 5]. The FHHM constructs a joint probability density such that each individual

appliance is marginally a simple hidden Markov model, where the hidden states being predicted are whether the appliance is on or off. In terms of modeling accuracy, FHHM has proven itself to be state-of-the-art [3]. However, it too is an exponential-time algorithm, and as of now, does not appear feasible on a large scale.

For this reason, we turn our attention to alternative approaches to energy disaggregation. In 2010, a study by Kolter, Batra, and Ng showed that a modified sparse coding methodology produced accurate disaggregation of whole-building energy signals [6]. Sparse coding first learns a dictionary of *atoms* for each appliance, essentially decomposed time series whose span includes the original signals for that appliance. The whole-building signal is then decomposed into linear combinations of these atoms following a regularized optimization procedure. While this formulation is unsupervised, the authors developed a discriminative training step which led to improved results [6].

Despite the promising finding, few subsequent papers have applied sparse coding [7]. In addition, the work centered solely on a dataset that is not publicly available. For this reason, we decided to implement their methodology and test its generalizability to the REDD dataset. We compare the result with benchmarks that we developed based on autoregressive time series models.

3 Dataset

REDD

We used the REDD dataset for our analyses, available from [8]. The data consists of six houses measured over a period of around 4-6 weeks. Each house has a building energy meter as well as separate appliance and circuit submeters. While the dataset is measured in a low frequency (1Hz) and a high frequency (15kHz), we only used the low frequency data because such data would be more widely collectable for future datasets.

The data was downloaded in a raw form and processed using tools from NILMTK, a Python library designed for the processing of energy disaggregation data [3]. In this processing stage, we aggregated the data into equally spaced intervals and saved the outputs as csv files. Missing values were imputed linearly. Note that our algorithm reference paper [6] used a time-scale of 1 hr, but their dataset spanned more than two years. For our much shorter dataset, we decided to aggregate in 300-second intervals using the mean. This resulted in around $T = 10000$ separate observations. Each house contains a varying amount of submeters. We sorted the appliances by energy usage and selected the top six for each house, so that we could train our models over a consistent set of devices.

Because we anticipated heterogeneity between houses, we decided to train the models separately for each house. Thus for buildings 2-6, the period from Apr. 19, 2011 - May 11, 2011 was designated as the training set. The period from May 12, 2011 to the end was used as the validation set for the models fitted specifically on the training set from the same building. Then once hyperparameters were tuned over the validation set, building 1 was saved as the final test set. The final configuration of the models were trained over the period Apr. 19, 2011 - May 11, 2011 on building 1, and then the period May 12, 2011 to the end was used as the final test set. Thus no tuning was done on building 1, which functions as the holdout set.

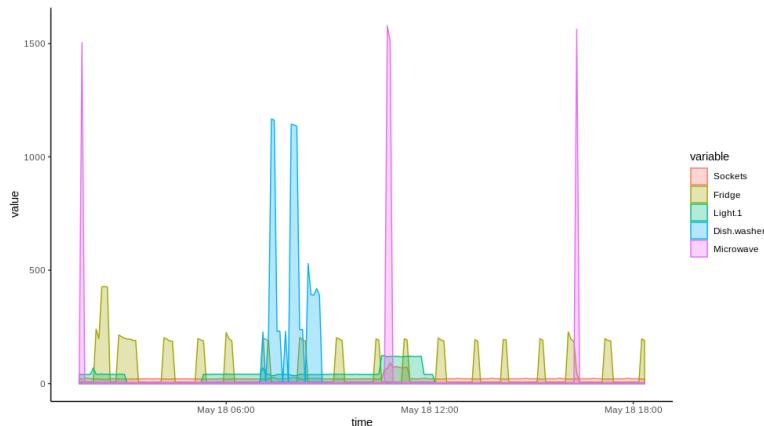


Figure 1: True energy disaggregation from REDD building 1

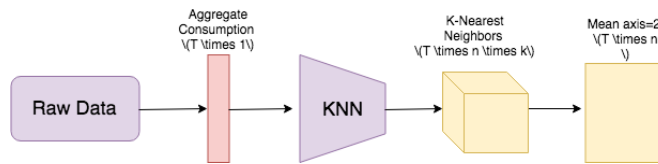
Weather

We explored if adding weather data would benefit our models. Since the REDD data is collected from the Greater Boston Area, we obtained historical weather data from Boston, MA over the period Apr. 19, 2011 - June 30, 2011 from the National Centers for Environmental Information [9]. We collected three variables: temperature, precipitation, and air pressure, anticipating that temperature and cloud cover would be associated with patterns of climate-control and lighting appliances. We aggregated the data to follow the same intervals as the REDD data and imputed missing values linearly.

4 Methods

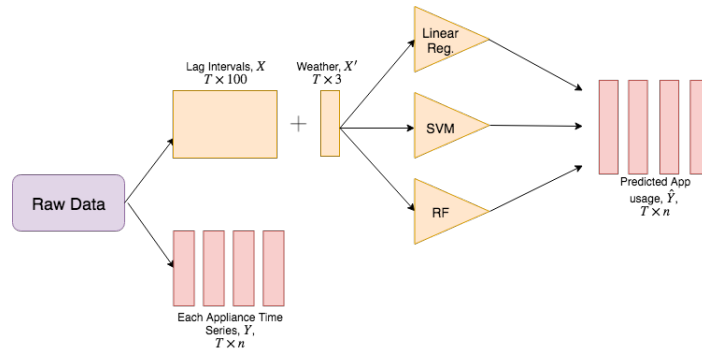
Suppose that for each house, there are T observations of n appliances, which we are trying to predict. Thus in this formulation, we have n different target variables, represented as $Y \in \mathbb{R}^{T \times n}$, with Y_i representing each appliance. Following the procedure of [6], we took the sum of the appliance energies over each time step as our predictor variable $X \in \mathbb{R}^{T \times 1}$, the whole-home energy time series. We also have the weather matrix $X' \in \mathbb{R}^{T \times 3}$.

Baselines



We formulated the baseline as n separate regressions $Y_i \sim X$. We first attempted a KNN model where we predict the continuous-valued kWh usage of each appliance at each time step by finding the k nearest observations that are close to the aggregated energy usage at t_i . Find the k nearest neighbors over all time steps T gives us a matrix of size $T \times n \times k$, where n is the number of appliances from the training data. We then reduce along the second dimension by taking the mean of the k observations to attain a prediction for our model. We experimented with $k = 3, 5, 7$ as k is ideally odd and tuned our choice of k by tracking accuracy on a validation set which consisted of 10% of our training set (homes 2 - 6), with home 1 as the testing home.

Time series models



We again run n separate regressions $Y_i \sim X$. However, instead of a single predictor variable, we constructed a 100-lag matrix $X^* \in \mathbb{R}^{T \times 100}$, where the first column is just X , the second is X lagged by one observation, etc., since clearly a wide range of past values of the total home time series could be informative. We added weather data to the above models simply by concatenating the matrices to get $[X^*, X']$. We constructed three models of increasing complexity:

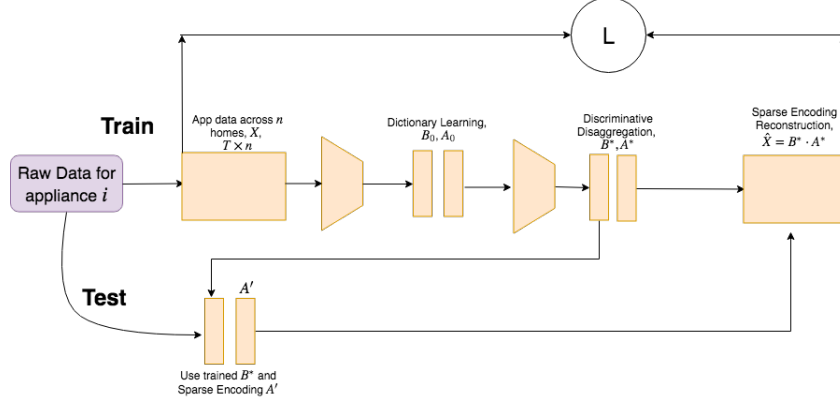
1. Ridge regression: This minimizes $(Y - X\beta)^T(Y - X\beta) + \lambda\|\beta\|^2$. Because lagged predictors are highly autocorrelated, simple least squares regression would be unstable. Thus we use L_2 -regularization to stabilize the model. For each model, we fit λ over the training set using 10-fold cross-validation.
2. Support vector regression: In order to fit a higher-dimensional feature-space of the predictors using the optimal margin classifier, we used the support vector machine, with rbf kernel set at $C = 1.0$, $\epsilon = 0.1$. Specifically, we use the Gaussian kernel

$$k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

to map $x_i \cdot x_j$ to $\phi(x_i) \cdot \phi(x_j)$, where ϕ can be infinite-dimensional output.

3. Random forest: Although the support vector machine can fit infinite-dimensional feature-space, we used a random forest model because it is non-parametric and an ensemble, and capable of learning discontinuous patterns of features. For example, plateaus or troughs of the lag matrix over certain magnitude or duration could be highly informative of certain appliances. 100 separate decision trees were fit on the data, and the mean of the trees was used as the prediction. KNN, Ridge, SVM, and random forest were fit using scikit-learn [10].

Supervised sparse coding



Modeling the problem of energy disaggregation as a sum of components where we learn a mapping $\mathcal{R} \implies \mathcal{R}^n$, we are reminded of ICA. Reading into literature, we find that Ng. et. al. (2009) apply dictionary and sparse encoding matrices to attempt source separation where we learn a set of k basis functions represented by $B_i \in \mathcal{R}^{T \times k}$ for each of $i = 1, \dots, n$ class of appliance. We now, instead, design an input matrix to the model as $X_i \in \mathcal{R}^{T \times m}$ where m is the number of homes. The idea is to learn a dictionary representation that captures the signature of each appliance such that $X_i \approx B_i A_i$, where $A_i \in \mathcal{R}^{k \times m}$ are activations of the bases. Thus, we are learning A_i^* and B_i^* that minimize the objective given by:

$$\min_{A_i, B_i} \frac{1}{2} \|X_i - B_i A_i\|_F^2 + \lambda \sum_{p,q} (A_i)_{pq}$$

where we enforce sparsity on activation matrix A through regularization and also subject the L2 norm of $\|B_i\|_2 \leq 1$ for each appliance $i \in 1..n$. In training, we thus perform 2 optimization steps much similar to k-means where we first initialize A and B with small positive values and for each appliance, first find the optimal A_i and then treat A_i as constant to optimize for B_i subject to the constraints. At testing time, we store our trained dictionary of basis functions for each appliance B^* and recompute the optimization for activations. We can then predict for each appliance using the dot product, $B_i^* A_i^*$.

5 Results and Discussion

Our primary evaluation metric was the fraction of total energy assigned correctly, defined as:

$$\sum_n \min \left(\frac{\sum_n y_{n,t}}{\sum_{n,t} y_{n,t}}, \frac{\sum_n \hat{y}_{n,t}}{\sum_{n,t} \hat{y}_{n,t}} \right)$$

where $y_{n,t}$ is true value for appliance n at time t , and $\hat{y}_{n,t}$ is the predicted value. The value ranges from 0 to 1, with 1 being perfect assignment.

We tuned the hyperparameters for KNN and ridge regression over the validation sets from houses 2-6 (see Methods for details). The other methods did not require any tuning. Refer to Table 1 for our model performances, and the following figures for the plotted fractions. As expected, KNN showed the poorest performance. Our time series models performed surprisingly well, even the ridge regression, which only models linear combinations of the input features. The SVM did not significantly outperform the linear model. However, the random forest showed large improvement and performs comparably with the sparse coding model. This suggests that nonparametric feature modeling may be a powerful tool for energy disaggregation.

Interestingly, performance on the test set was high, especially compared to the dev set, which we believe is due to the data from building 1 being relatively more well-behaved compared to the other buildings. Thus, with or without fine-tuning,

Table 1: Results for models

Model	Train	Dev	Test
KNN	0.57	0.46	0.49
Ridge	0.85	0.80	0.83
SVM	0.86	0.81	0.84
RF	0.95	0.85	0.89
RF + weather	0.95	0.84	0.89
Sparse coding	0.92	0.90	0.91

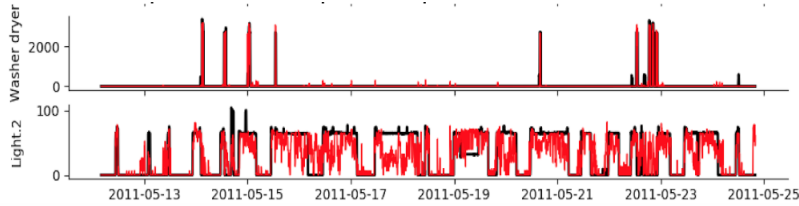
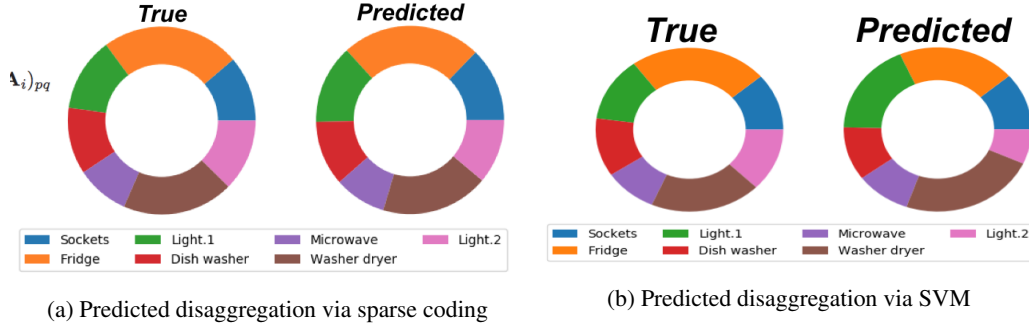


Figure 3: Random forest prediction for lights and washer / dryer

models tended to fit better on building 1. As expected due to the complex nonparametric modeling, the random forest achieves extremely accurate results over the training set but drops significantly out-of-sample, while the other time series models show less of a drop. In addition, we found that weather did not seem to influence prediction performance. Finally, sparse coding proved to be a strong-performing model, being roughly comparable to random forest.

Our finding lends support to the success of the modified sparse coding method. As its pie chart shows, it predicts each device class roughly equally well. On the other hand, the SVM is less accurate, overpredicting socket usage for example. Sparse coding also shows robustness to overfitting, which may be related to the effect of regularization, which lowers variance while introducing some bias. It is interesting that random forest simply on a matrix of lagged time series data proved to be one of the top-performing models. That may be because it is an ensemble, which increases robustness and reduces variance.

6 Conclusion

Here we have explored several algorithms for disaggregating whole home energy consumption into individual appliances. Starting with low accuracy on our baseline KNN model, we have been able to find that non-linear models such as Random Forests and Sparse coding performed the best on the prediction task with 0.91 and 0.89 accuracy on our testing home. We also experiment with data augmentation using limited amount of weather data and we hypothesize that adding a larger amount of metadata will help increase model accuracy. Seeing the relationship in previous time steps to current is non-linear, we hope to explore a few additional approaches including convolution and possibly gated recurrent networks with attention. We also note that our model may not be general training with only 6 homes in a single city, and thus hope to train our systems with a larger and more representative dataset.

7 Contributions

SS and FL contributed equally to all sections of the project. In particular, Samir worked on building the sparse coding, KNN and experimental neural networks while Fred worked on building the data processing and visualization, random forest, Multi-task + Ridge Regression, SVM. Both worked on data preprocessing and augmentation as well as visualization.

References

- [1] J Zico Kolter and Matthew J Johnson. Redd: A public data set for energy disaggregation research. In *Workshop on Data Mining Applications in Sustainability (SIGKDD)*, San Diego, CA, volume 25, pages 59–62. Citeseer, 2011.
- [2] George William Hart. Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 80(12):1870–1891, 1992.
- [3] Nipun Batra, Jack Kelly, Oliver Parson, Haimonti Dutta, William Knottenbelt, Alex Rogers, Amarjeet Singh, and Mani Srivastava. Nilmtk: an open source toolkit for non-intrusive load monitoring. In *Proceedings of the 5th international conference on Future energy systems*, pages 265–276. ACM, 2014.
- [4] Hyungsul Kim, Manish Marwah, Martin Arlitt, Geoff Lyon, and Jiawei Han. Unsupervised disaggregation of low frequency power measurements. In *Proceedings of the 2011 SIAM international conference on data mining*, pages 747–758. SIAM, 2011.
- [5] J Zico Kolter and Tommi Jaakkola. Approximate inference in additive factorial hmms with application to energy disaggregation. In *Artificial Intelligence and Statistics*, pages 1472–1482, 2012.
- [6] J Zico Kolter, Siddharth Batra, and Andrew Y Ng. Energy disaggregation via discriminative sparse coding. In *Advances in Neural Information Processing Systems*, pages 1153–1161, 2010.
- [7] Ehsan Elhamifar and Shankar Sastry. Energy disaggregation via learning powerlets and sparse coding. In *AAAI*, pages 629–635, 2015.
- [8] J. Zico Kolter. The reference energy disaggregation data set.
- [9] National Oceanic and Atmospheric Administration. National centers for environmental information. <https://www.ncdc.noaa.gov/>.
- [10] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.