Joan Wang

CMSC 123 Spring 2014

**REDD Energy Disaggregation Prototype Write-up**

In our initial project proposal, we hoped to have a prototype that we could compare against the REDD FHMM model. We wanted to use the same process and see how accurate our model is, in order to explore the data before trying methods not used by REDD.

For this prototype, we began by researching HMM and FHMM models and their uses, creating two powerpoints to record our research: Joan\_HMM\_Interpretation.pptx and REDD FHMM.pptx. We saw the process for creating an HMM model using the Baum-Welch (EM) algorithm, by building the alpha and beta matrices for the model, adjusting variable, and reiterating the process until we meet a minimum difference from the target log probability. We found the way to use cross-products to translate a Factorial Hidden Markov Model, which we want to use since it incorporates multiple states, into a Hidden Markov Model form, so that we can use the EM algorithm to build the model.

We decided to test our prototype on one house first, House 2. The data, after downloading from the REDD site, was fairly easy to clean—it was in text form, and we transposed the data into csv format. The entries of the data were fairly clear: it was divided by house folders, which contained a file for each circuit that included timestamps and the wattage at that time stamp. We used the scikit-learn library from python, which includes functions for the hmm model, though I’m considering writing code from scratch later to get a better understanding of the FHMM. To build the model in a big data method, we used mapreduce via python’s mrjob. I took advantage of mrjobs ability to have multiple mapper and reducer functions, and used the map to output HMMs for each circuit in the house in parallel, and then combined these HMMs into a FHMM for the house in the reducer functions. In order to run models for multiple houses, I would try to wrap these the current code into another mapper function that outputs the FHMM for each house, and then determine the average accuracy of all the FHMMs in the reducer function to test the accuracy of the model.

During the process of building the model, we found that the data itself was not hard to deal with, though this was probably because we were only able to access 3GB of the data, so manipulating the data was not too difficult. The most important things we’ve learned so far is in terms of the algorithm, which if fairly different from what we’ve done in class so far. In terms of the analysis of the data, since we’re just replicating the results of the REDD paper for now, we have not yet found anything about the data that we have not yet read from the paper.

For the final product, I plan on building code that would take the FHMM and show its output in a visual way, as well as show its accuracy as compared to the model in the REDD paper. I also plan on building other models using K-nearest neighbors, and sparse code algorithms, and then comparing the accuracy of these to the baseline FHMM accuracy.