Energy Aggregation using Product of HMMs

Megha Gupta[1], Haimonti Dutta[2] * Ullas Nambiar[3], and Amarjeet Singh[4]

[1] IIIT Delhi, India,
meghag@iiitd.ac.in,
[2] State University of New York, Buffalo,
haimonti@buffalo.edu
[3] EMC, India
Ullas.Nambiar@emc.com
[4] IIIT Delhi, India
amarjeet@iiitd.ac.in

Abstract. Real time spatio-temporal energy consumption data is captured by large scale deployment of smart meters. Data from these meters are usually sent to a base station (BS) where they are aggregated for analytics. Each BS aggregates the load derived from all the meters connected to that station. The readings received at the BS are adhoc and usually not synchronized in time. Different smart meters can send data points when they are collected resulting in inconsistent data including aggregating non-aligned time stamped readings, readings with missing values, repeated values, meter reset readings. We address the problem of learning from disparate data streams (with inconsistencies) by modelling streams as HMMs and the process of aggregating data at the BS as a Product of HMMs. This enables us to perform load forecasting using machine learning techniques. Empirical results are presented on two data sets - Reference Energy Disaggregation Data (REDD) and energy consumption data collected from faculty housing at IIIT-Delhi. The results show that this technique performs the best by combining via product, all the HMMs (corresponding to each data stream) with binary states (on, off or standby) and training time linear to the number of HMMs.

Keywords: We would like to encourage you to list your keywords within the abstract section

1 Introduction

Smart meters consisting of real time sensors, power outage notifications and power quality monitoring are widely used today. These meters provide a host of benefits like energy efficiency and savings, improved retail competition, better demand response actions, improved tariffs, lower bills due to better customer feedback, accurate billing, less environmental pollution, etc. [19] They generate

* The author is also affiliated to the Institute of Data Science and Engineering (IDSE), Columbia University and is an adjunct professor at IIIT-Delhi.

huge amount of time series data which can be used for gaining meaningful insights through analytics. They can measure site specific information and also help agencies set different electricity prices for consumption based on the time of the day, seasons, holidays, etc. Based on the data collected from smart meters, a feedback sent to the customers by the utilities that can help consumers better manage their resources. McKerracher et al. [23] show that by providing real time feedback, consumers can reduce the consumption by 3-5%.

In recent years, machine learning has been applied to the problem of energy consumption and demand forecasting analysis. The role of the machine learning algorithm is to study the sensor data and provide alerts and warnings when anomalous behaviour occurs or to inform (and remind) customers when certain activities were performed, which rooms they occupied, and what appliances they used most frequently during that period. This information can be transmitted to customers in timely fashion via phone, email or the Internet. Chicco et al. [7] compared several clustering techniques (such as hierarchical, KMeans) and observed that the hierarchical clustering and modified follow-the-leader perform best among the rest K-Means, fuzzy K-Means to group customers with similar electrical behaviour [22]. Wijaya et al. [32] used classifiers like random forest, decision trees (J48), logistic and naive bayes to identify customers with similar electricity consumption profiles. Related problems involve study of trends of electricity consumption (steadily increasing, decreasing, cyclic, seasonal) and sudden anomalous behaviour (sudden peaks or drops on consumption) for individual homes and across the community [9].

In this paper, we use Hidden Markov models (HMMs) to analyse the time series energy data. We model the data stream from each source as a HMM with its states represented as ON/OFF. For N sources, there are N HMMs and the total number of states collectively are 2^N. The observations represent the energy consumed in a particular state. These observations are recorded at different time scales for different sources.
In order to aggregate the data from all the different sources, we build a machine learning model using products of HMMs (PoHMMs) and apply it to the energy aggregation problem. There are many reasons why the product model constructed from many HMMs is appropriate. First, in a high-dimensional space each model constraints a different subset of dimensions but their product constraints all of the dimensions. Second, HMMs alone are not efficient at capturing long range structure in time series [31] – in contrast to PoHMMs [5] allow each model to remember a different piece of information about the past. Two different proof of concepts are presented – first one on the REDD [5] data set and the other one on real data collected at the faculty housing in India.

Organization: This paper is organized as follows: Section 2 examines related work on data analytics on aggregated data of smart meters; Section 3 provides a review of products of Hidden Markov Models (HMMs) and how they relate to our application. The two proofs of concepts are introduced in Section 4 illustrat-

[5] http://redd.csail.mit.edu/

ing the effectiveness of the use of product of HMMs in the energy aggregation problem. Finally, Section 5 concludes the work.

2 Related Work

In this section, we describe work that uses ensemble learning techniques and non-ensemble learning techniques to solve problems in energy domain.

2.1 Non-ensemble based learning techniques

Energy Aggregation In wireless sensor networks, energy data aggregation is a method of combining data from different sources such that several unreliable data measurements combine to produce a more accurate signal by enhancing the common signal and reducing the uncorrelated noise. As the sensor network generates lot of data for the end user to process, there are automated methods employed to aggregate data. This data fusion is generally known as data aggregation which combines the data into a set of meaningful information [15]. The sensor nodes are organised in a tree structure, called aggregation tree. The leaves of this tree are the sensor devices, the internal nodes are the aggregator devices that takes the data from the leaves, aggregates it and sends it to its parent node which is the root of the tree.

The main objective of data aggregation is to reduce the unnecessary information thereby reducing the network traffic and improving the privacy of the customers from internal and external entities by keeping only the necessary information [30].

Energy Disaggregation The process in which the whole building energy (aggregated) signal is separated into appliance level energy (disaggregated) for a variety of reasons like residential energy reductions, program evaluation, targeted marketing, etc. Several studies have been done in this regard, one of the unsupervised desegregation method [18] that outperforms other unsupervised disaggregation methods is conditional factorial hidden semi-Markov model. This model when integrated with other features, accurately represents the individual appliance energy consumption. Kolter et al. [21] exploits the additive structure of the FHMM to develop a convex formulation of approximate inference algorithm that achieves state-of-the-art performance in energy disaggregation problem.

Load Forecasting Electrical load forecasting refers to the projection of electrical load required in a certain geographical area with the use of previous electrical load usage in the same area. It is extremely important for efficient power system planning and operation, energy purchasing and generation, load switching, infrastructure development. It encompasses various factors like, historical load, weather data, population, energy supply and price, time of the year, etc. It is usually divided into three categories, short-term forecasts (one hour to one week) , medium-term forecasts (one week to one year) and long-term forecasts (more than a year). In short term load forecast, [2] and [6] used a three layer feed forward artificial neural network to predict daily load profiles. In a paper by

[8], nonlinear autoregressive integrated neural network was used to predict daily load consumption. In medium term load forecasts, the author forecasts [11] the monthly load through knowledge based activities from the output of the ANN based stage providing yearly energy predictions. Similarly, in [3], time lagged feedforward neural network is used to do monthly forecasting on the basis of historical series of electrical load, economic and demographic variables. Also, the authors from covenant university, [26] performed load forecasting of their own educational institute using the models based on linear, compound growth and cubic methods of regression analysis. In long term load forecasting, study done by [10] resulted in showing that the models based on regression analysis did not give very accurate predictions as compared to fuzzy neural network which performed better due to better handling with non linear systems. Another work [34] uses support vector regression to derive non linear relationship between load and economic factors like GDP for long term forecasting in developing countries.
Customer Segmentation The identification of consumer profiles that show similar behaviour in energy consumption. This analysis is useful in various ways, like demand response system, intelligent distribution channel. The author [33] segments the customers based on contextual dimensions like location, seasons, weather patterns, holidays, etc which help with various higher level applications like usage-specific tariff structure, theft detection, etc. In [1], author proposes to infer occupancy states from consumption time series data by using HMM framework. They investigate the effectiveness of HMM and model based cluster analysis in producing meaningful features of the classification. This work suggests the dynamics of time series as captured by HMM analysis can be valuable.

2.2 Ensemble based learning techniques

Ensemble learning is a method where multiple learners are trained to solve the same problem. It constructs a set of hypothesis and combines them to generate the final result.

Prediction with expert advice A study done by [27], proposed a Pattern Forecasting Ensemble Model (PFEM) comprising of five forecasting models using different clustering techniques, like k-means model, self-organising map model, hierarchical clustering model, k-medoids model and fuzzy c-means model. They have showed that on three real-world dataset, their proposed ensemble model outperformed all the five individual model in case of day ahead electricity demand prediction. Another study [12] highlighted the importance of regularised negative correlation learning ensemble methodology on the problem of energy load hourly prediction. This method tried to overcome the problem of variability in neural network due to high sensitivitiness to the initial conditions. As this method combines the outputs of several neural networks, it achieves a marked reduction in error after introducing external data.
An extension of HMMs, called Factorial Hidden Markov Model (FHMM) [14] is a class of ensemble based learning models that addresses the need for distributed

hidden states in HMMs. But by being a directed model, when conditioned on the observed sequence, the hidden state chains become independent making the inference easy but learning more complex. But this approach has proved to be very inefficient in high dimensional spaces. To model a complicated, high-dimensional data distributions, an approach called mixture of gaussians is widely used. In this method, each simple model which is a gaussian is combined using a weighted arithmetic mean of individual distributions. Such mixtures of tractable models can easily fit to data using expectation-maximization (EM) and are more powerful than their individual component. So, a different way of combining these distributions is by multiplying them together and then renormalizing them. In our paper, we deal with the problem of energy aggregation using ensemble learning model. Each HMM is used to represent a state of an appliance. An appliance can have states like ON or OFF. The combination of the outputs from each of these HMM models gives us our ensemble based learning model, Product of Hidden Markov Model (PoHMM) [16]. This learning technique outputs the probability distribution by combining the outputs from several simpler distributions. It allows each model to make a decision on the basis of few dimensions.

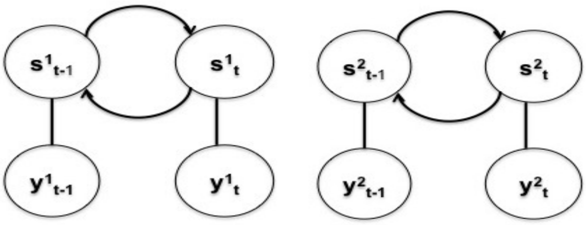


Fig. 1. HMM S^1 and S^2

3 Review

A Hidden Markov Model (HMM) is a statistical Markov model that represents the probability distribution over a sequence of observations [13]. They are found useful in applications like speech [25], handwriting, gesture recognition, part-of-speech tagging, bioinformatics, etc. It has two properties, first, the observation at time t, y_t is generated by a process whose state at time t, s_t is hidden from the observer and second, is that this hidden state process satisfies Markov property which states that given the value at state s_{t-1}, the value at current state s_t is independent of all the states prior to $t-1$. The subscripts i and superscripts j indicate the model at ith time and the jth HMM. The state space of the HMM is discrete, that is a state can take 2 values denoted by ON and OFF. The observed values represent the aggregated load/energy collected from different data streams at time t. In order to define probability distribution over the sequence

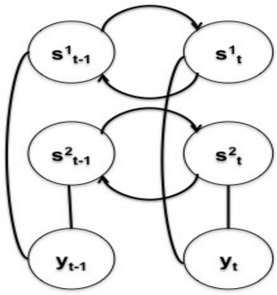


Fig. 2. Product of HMMs, P = S^1 x S^2

of observation, it is important to define probability distribution over the initial state P(s_1), the transition probability P($s_t|s_{t-1}$) and the observed probability P($y_t|s_t$) where y_t is the observation at time t.
Following a notation in [25], HMM is composed of a 3-tuple {A, B, π } where A is the transition probability, B is the observed probability and π is the initial state probability. HMMs solve three fundamental problems: 1. Given the model λ = {A, B, π }, and observation sequence Y = {y_1,...,y_T}, how do we efficiently compute the probability of the sequence of observations given the model, that is P($Y|\lambda$). 2. Given the model λ and observation sequence Y, what is the underlying state sequence {s_1,...,s_T} that best explains the observations. 3. Given the observation Y and state space sequence S, how do we need to adjust the parameters so as to find the model λ that maximises P($Y|\lambda$).
In this paper, we deal with the third problem as it involves learning parameters by training the model with the historical data and then using these parameters to predict the future observations. The figure 2.2 shows the HMM S^1 and S^2 generated by a data stream.

3.1 Product of HMMs

PoHMM is a model that combines several HMMs by multiplying their individual distribution together and then renormalizing them. Its representation includes both directed and undirected links where the hidden states are causally connected to the other hidden states but non causally related to the visible states. This causes different conditional independence relationships among the variables in graphical model. The figure 2 is a product of two HMMs P = S^1 x S^2 where the superscript in S^1 indicates the kth HMM. The number of states in the PoHMM is the product of states in S^1 and S^2 which is 4 in our case. The connections formed in the P depend on the links in the multiplying HMMs.

3.2 Training the model by minimising contrastive divergence

To fit the model to the data, we need to maximize the likelihood of the dataset or minimise the Kullback-Liebler divergence between the data distribution, P^0

(data distribution at time 0) and the equilibrium distribution over the visible variables, P_θ^∞ (fantasy data) which is obtained after prolonged Gibbs sampling as shown in equation 1.

$$P^0||P_\theta^\infty = \sum_d P^0(d)logP^0(d) - \sum_d P^0(d)logP_\theta^\infty(d) \tag{1}$$

$$P^0||P_\theta^\infty = -H(P^0) - \langle logP_\theta^\infty \rangle_{P^0} \tag{2}$$

where || represents Kullback-Leibler divergence, d is the data vector in discrete space, θ_m is all the parameters of individual model m, P^0 is the data distribution at time 0, H(P^0) represents the entropy which is ignored during optimisation as P^0 does not depend on the parameters of the model, angle brackets denote the expectation over the distribution specified by the subscript. In Gibbs sampling, each variable draws a sample from its posterior distribution given the current states of the other variables. The hidden states of all the models are conditionally independent given the data and hence can be parallel updated as shown in Figure 3. At time t=0, the observed variables represent a data vector, d and the hidden variables, s of all the models are updated in parallel with samples from their posterior distribution given the observed variables, y. At time 1, the visible variables are updated to generate a reconstruction of the original data vector from the hidden variables and the hidden variables are again updated simultaneously. This prolonged sampling helps the Markov chain to converge to the equilibrium distribution which helps to attain the unbiased estimate of the gradient of the PoHMMs. But since the samples from the equilibrium state have high variance as they come from the entire model's distribution, it poses a difficulty in determining the estimate the derivative. Therefore, the optimisation is performed on the different objective function called contrastive divergence, defined in equation 4. Contrastive divergence is the difference between $P^0||P_\theta^\infty$ and $P_\theta^1||P_\theta^\infty$ where P_θ^1 is the distribution over the one-step reconstruction of the data vectors generated by one full step of Gibbs sampling. The intuition behind using contrastive divergence is to leave the initial distribution P^0 over the visible variables unaltered and also the intractable expectation over P_θ^∞ gets cancelled out. Instead of comparing the initial and final derivatives, P^0 and P_θ^∞, the Markov chain is run for one full step and the parameters are updated to avoid the chain to wander away from the initial distribution on the first step. As P^1 is a step closer to P^∞ which guarantees that $P^0||P_\theta^\infty$ will always exceed $P_\theta^1||P_\theta^\infty$ ensuring a non negative value unless $P^0 = P_\theta^1$. If $P^0 = P_\theta^1$, then it implies that the chain is already in an equilibrium state, that is $P^0 = P_\theta^\infty$ hence making the value of contrastive divergence as 0.

$$\begin{aligned} -\frac{\partial}{\partial\theta_m}(P^0||P_\theta^\infty - P_\theta^1||P_\theta^\infty) = \langle\frac{\partial logf_{\theta_m}}{\partial\theta_m}\rangle_{P^0} \\ -\langle\frac{\partial logf_{\theta_m}}{\partial\theta_m}\rangle_{P_\theta^1} + \frac{\partial P_\theta^1}{\partial\theta_m}\frac{\partial(P_\theta^1||P_\theta^\infty)}{\partial P_\theta^1} \end{aligned} \tag{3}$$

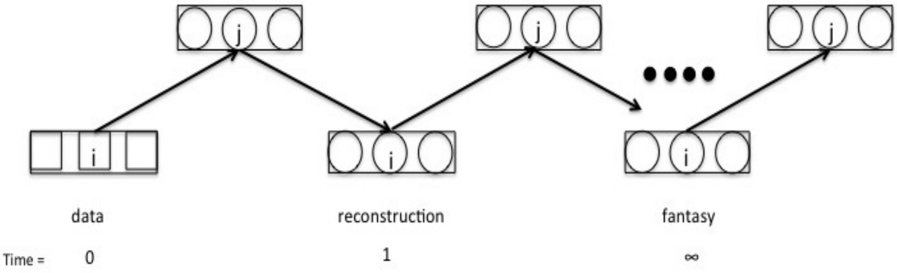


Fig. 3. Visualization of Gibbs sampling

where logf_{θ_m} is a random variable that can also be written as $f_m(D|\theta_m)$ where D being a random variable corresponding to the data. In equation 4, the first two terms on the right hand side are tractable as it is easy to sample from P^0 and P^1_θ but the third term represents the effect on $P^1_\theta||P^\infty_\theta$ of the change of the step reconstruction caused by the change in the θ_m. Extensive simulations show that it is small and rarely differs from the result of other two terms, hence can be safely ignored. Therefore in contrastive divergence, the parameters are learned according to the equation 4. To minimise the contrastive divergence by using a Markov chain that slowly mixes, we can use mixing techniques like weight decay that ensures that every possible visible vector has non zero probability given the latent variables.

$$\Delta\theta_m \propto \langle\frac{\partial log f_{\theta_m}}{\partial\theta_m}\rangle_{P^0} - \langle\frac{\partial log f_{\theta_m}}{\partial\theta_m}\rangle_{P^1_\theta} \tag{4}$$

The contrastive divergence algorithm for training the PoHMM has the following steps:

1. Each model's gradient $\frac{\partial}{\partial\theta_m}$ $P(Y|\theta_m)$ ($\{y_t\}_{t=1}^T$ = Y is the visible variable) is calculated on a data point using forward backward algorithm.
2. A sample for each model is taken from the posterior distribution of paths through state space.
3. At each time step, the distributions are multiplied and renormalized together to get the reconstruction distribution.
4. A sample from the reconstruction distribution is drawn at each time step to get a reconstructed sequence. Each model's is gradient is computed on the new sequence $P(\hat{Y}|\theta_m)$
5. Parameters are updated as per equation 4

3.3 Inference in PoHMM

The main feature of PoHMMs is its undirected graphical modelling with no direct connection among the latent variables (S^1_t and S^2_t) as they only interact indirectly via observed variables (Y_t). The hidden variables all the experts are rendered independent when conditioned on visible variables. So, if the inference in each of the constituent model is tractable then the inference in the product is also tractable. To generate a data point in this model, all the models in PoHMMs generate an observation and if they all generated the same point then it is accepted else they again generate an observation until all the models agree to it. Therefore all the models have some influence over the generated data. So, the inference determines the the probability that all the models would have taken in order to generate the given observation.

4 Applications using Product of HMM

Aim: In this section we demonstrate how PoHMMs can be used to model data streams and perform load forecasting. Proof-of-concepts are provided on two data sets - REDD[6] and on the energy data collected from faculty housing at IIIT Delhi.

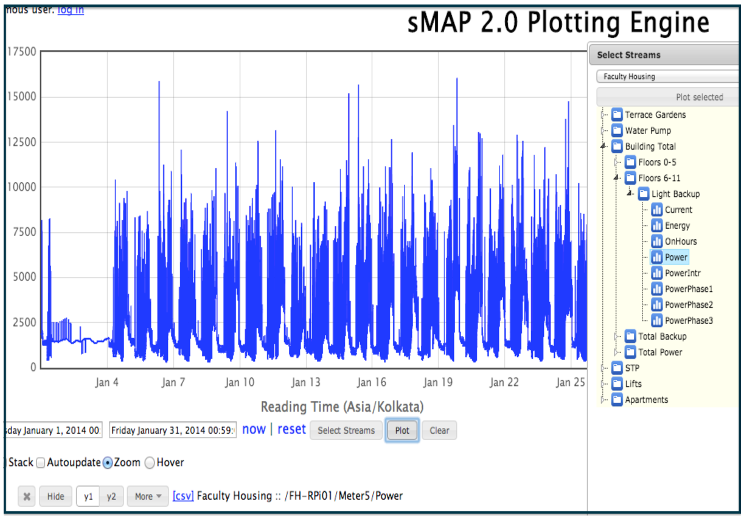
4.1 Data Description

- **Dataset 1:** The Reference Energy Disaggregation Data Set (REDD) contains power consumption data from real homes, for the whole house as well as for each individual circuit in the house (labeled by the main type of appliance on that circuit). The REDD data set contains two main types of home electricity data: high-frequency current/voltage waveform data of the two power mains (as well as the voltage signal for a single phase), and lower-frequency power data including the mains and individual, labeled circuits in the house. Experiments reported here use 'house 2' data from REDD. This dataset has 318759 records and 2 columns. We randomly sample 300 records for our initial experiment.

  Implementation Details The implementation of the product of hidden Markov model is obtained from Iain Murray's website[7]. It implements the technique described in paper [16].
- **Dataset 2:** This data represents the energy consumed by the IIIT Delhi's faculty housing building. As a part of research, a team from the institute has installed various temperature, light and motion sensors to perform real world studies and to analyse user preferences for energy conservation. For our analysis, we selected 1 month's historical data ranging from 01-01-2014, 00:01 hours to 31-01-2014, 23:59 hours. The two smart meters installed, captured data from all the 12 floors. The first meter records readings from ground to 5th floor generating one stream of data and the second meter generates a stream from 6th to 11th. Both these streams are aggregated to obtain the

[6] http://redd.csail.mit.edu/

[7] http://homepages.inf.ed.ac.uk/imurray2/code/



sday January 1, 2014 00 | Friday January 31, 2014 00:59:

Stack □ Autoupdate ● Zoom ○ Hover

[csv] Faculty Housing :: /FH-RPi01/Meter5/Power

Fig. 4. Screen shot of the tool that is used to download the energy data of the institute collected over the period of time.

aggregated load of the housing building. The dataset includes timestamp and power consumed in watts. There are 84133 records in this dataset. We also have the total energy consumed by the faculty housing building which would serve as the ground truth to compare the aggregated load using PoHMMs.

4.2 Problem Formulation

The disparate energy data streams collects readings at different time scales. Each of the data stream is modelled as a HMM with cardinality 2, that is either ON or OFF. The process of aggregating the energy data from different data streams is modelled through PoHMMs.

Each energy data stream is used to train the model, till the time the objective function, that is contrastive divergence reaches a threshold value. Once the model is trained from a randomly sampled data stream, the parameters learned by the model are provided to the randomly sampled test set (total energy consumed data) to obtain the conditional probability distribution of the gaussians given the data. Similarly, all the data streams are used for training the model, and the parameters learned are then applied on the test set to obtain the conditional probability of the gaussians given the data. After all the data streams are used to obtain the probability distribution, we use the data stream that correspond to the total energy consumed from the house/ building to train the model and hence obtain the probability distribution P of the gaussians. These probability distributions are then compared with the product of the probability distributions Q obtained from the individual data streams. The evaluation of how well the learning has taken place is done by using a Kullback-Leibler divergence. KL divergence of two probability distributions P and Q, $D_{KL}(\mathrm{P}\|\mathrm{Q})$ is the measure of information lost when Q is used to approximate P.

Samples	$KLDiv$	$Iterations$	$T(sec)$
300	2.4864	18600	186.212 ±9.087
500	0.6761	10200	106.564 ±10.046
1000	1.1088	11200	158.521 ±1.97
1500	3.8829	5300	92.896 ±8.075
2000	1.8686	6900	130.98 ±1.932
2500	0.4733	9900	215.563 ± 2.471
3000	2.8204	11000	258.213 ±1.918
3500	1.2332	7900	204.661 ±1.713
4000	0.8959	10400	292.666 ±0.619
4500	1.1118	7200	222.558 ±1.967
8000	6.392	8100	381.635 ±2.952
10000	8.276	10500	887.932 ±13.824
15000	0.7201	9400	1368.514 ±13.605

Table 1. Effect of varying samples on KL div and time for REDD

4.3 Empirical Results

In both the datasets, experiments are performed by tuning some parameters and keeping some fixed. In case of REDD, Tables 1, 2 and 3 show the effect of varying data samples, threshold and no. of appliances on the KL divergence. In Table 1, data samples are randomly selected from the dataset, KL divergence is chosen are the performance metric, iterations are performed to obtain the samples from the equilibrium distribution by using Gibbs sampler on the hidden and visible variables and the average time taken (sec) by these iterations are shown in the last column. In case of faculty housing dataset, Tables 4 and 5 show the effect of varying data samples and no. of HMMs on the KL divergence. As per table 1 and 4, the relation between no. of samples and KL divergence was observed, the best performance was attained when 2500 and 300 samples were randomly taken from the REDD and faculty housing dataset respectively. In table 2, there is an indirect relationship of threshold value and KL divergence, that is lower the threshold, higher the KL divergence in contrast to the direct relationship between the threshold value and the time taken or the iterations required. Table 3 and 5 show that when all the appliances in the home are used in PoHMMs the performance is at its best. The value of KL divergence is minimum when the energy data from all the appliances are aggregated and tested with total energy data.

Threshold	$KLDiv$	$T(sec)$	$Iterations$
.1	0.473	210.6 ±1.493	9900
.05	0.443	240.607±2.436	10900
.01	0.454	431.536 ±14.509	18000
.005	0.509	1167.243 ±43.412	49800

Table 2. Effect of varying min threshold on KL div and time for REDD

Appliances	$KLDiv$	$T(sec)$	$Iterations$
3	5.559	233.664 ±0.579	10700
4	0.188	465.634 ±5.275	19900
5	.432	338.416 ±3.988	13400
6	8.736	606.062 ±7.534	28100
7	5.054	411.457 ±10.051	17300
8	0.436	260.544 ±27.862	10700
9	0.15	474.579 ±14.619	20600

Table 3. Effect of varying appliances on KL div and time for REDD

Samples	$KLDiv$	$T(sec)$	$Iterations$
100	2.6219e-05	257	45100
300	1.9753e-05	222	43200
500	5.5493e-05	260	44800
700	3.2847e-05	249	44000
900	3.9486e-04	221	42600
1100	4.9274e-04	317	44700
1300	3.0425e-04	276	43100
1500	3.1128e-04	303	44400
2000	1.9192e-04	306	44400
2500	1.7122e-04	370	44100
3000	1.4686e-04	331	43300
3500	1.2663e-04	370	43200
4000	1.0793e-04	403	43200

Table 4. Effect of varying samples on KL div for housing data

HMMs	$KLDiv(e-05)$	$T(sec)$
2	0	219
3	1.9780	229
4	3.5897	217
5	1.9753	228
6	4.3488	238
7	4.9111	245
8	5.6564	241
9	5.4290	258
10	5.5163	267
12	4.4504	262
14	6.9006	296
16	6.8666	300
18	6.2872	313
20	5.3842	267
25	5.8970	326
30	5.9962	327
35	5.2716	346
40	5.0955	320

Table 5. Effect of varying HMMs on KL div and time for housing data

5 Conclusion

In this paper, we are solving the problem of load aggregation (energy consumption) for disparate energy data sources using the ensemble based learning technique called Product of HMMs. Several challenges are faced while computing the aggregated energy, few of which include non-aligned timestamp readings, missing values, meter reset readings. This technique produces the combined probability distributions of several simpler distributions and then renormalizes this output. The optimisation problem is to minimise the contrastive divergence between the two probability distributions of the data at time 0 and the data at time 1 (one-step reconstruction using gibbs sampling). The evaluation of the algorithm is done by computing the KL divergence between the product of the energy data distributions and the total energy data distribution. From the results shown in table 3 and 5, we can see that the algorithm has performed best when the number of appliances (REDD) reached the actual number of appliances (9) in the house and the number of data streams in case of housing data reached the actual number of streams (2) used in the 12 floor faculty housing data. Therefore, this signifies that this algorithm is applicable for this kind of application and can be used for load forecasting.

Acknowledgment

The authors would like to thank EMC^2 for supporting this research through grant number JRA092014-Q3-5.

References

1. Albert, A., Rajagopal, R.: Smart meter driven segmentation: What your consumption says about you. Power Systems, IEEE Transactions on 28(4), 4019–4030 (Nov 2013)
2. Bakirtzis, A., Petridis, V., Kiartzis, S., Alexiadis, M.: A neural network short term load forecasting model for the greek power system. Power Systems, IEEE Transactions on 11(2), 858–863 (May 1996)
3. Bassi, D., Olivares, O.: Medium term electric load forecasting using tlfn neural networks. International Journal of Computers, Communications & Control 1(2), 23–32 (2006)
4. Brown, A., Hinton, G.: Proceedings of artificial intelligence and statistics 2001. In: Products of Hidden Markov Models. No. GCNU TR 2000-008 (2001)
5. Brown, A.D.: Product model for sequences. Gatsby Unit (2001), `http://www.gatsby.ucl.ac.uk/publications/andy\_thesis\_final.pdf`
6. Chen, H., Canizares, C., Singh, A.: Ann-based short-term load forecasting in electricity markets. In: Power Engineering Society Winter Meeting, 2001. IEEE. vol. 2, pp. 411–415 vol.2 (2001)
7. Chicco, G., Napoli, R., Piglione, F.: Comparisons among clustering techniques for electricity customer classification. Power Systems, IEEE Transactions on 21(2), 933–940 (May 2006)

8. Chow, T., Leung, C.: Nonlinear autoregressive integrated neural network model for short-term load forecasting. Generation, Transmission and Distribution, IEE Proceedings- 143(5), 500–506 (Sep 1996)
9. Cook, D.J., Chen, C.: Teasing detailed home habits from aggregate energy consumption data. IEEE Smart Grid (February 2012), http://smartgrid.ieee.org/newsletter/february-2012/507-teasing-detailed-home-habits-from-aggregate-energy-consumption-data
10. Daneshi, H., Shahidehpour, M., Choobbari, A.: Long-term load forecasting in electricity market. In: Electro/Information Technology, 2008. EIT 2008. IEEE International Conference on. pp. 395–400 (May 2008)
11. Falvo, M., Lamedica, R., Pierazzo, S., Prudenzi, A.: A knowledge based system for medium term load forecasting. In: Transmission and Distribution Conference and Exhibition, 2005/2006 IEEE PES. pp. 1291–1295 (May 2006)
12. Felice, M.D., Yao, X.: Short-term load forecasting with neural network ensembles: A comparative study [application notes]. IEEE Comp. Int. Mag. 6(3), 47–56 (2011)
13. Ghahramani, Z.: Hidden markov models. In: World Scientific Series In Machine Perception And Artificial Intelligence Series, chap. An Introduction to Hidden Markov Models and Bayesian Networks, pp. 9–42. World Scientific Publishing Co., Inc., River Edge, NJ, USA (2002), http://dl.acm.org/citation.cfm?id=505741.505743
14. Ghahramani, Z., Jordan, M.I.: Factorial hidden markov models. Mach. Learn. 29(2-3), 245–273 (Nov 1997), http://dx.doi.org/10.1023/A:1007425814087
15. Heinzelman, W.R., Ch, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: System Sciences. pp. 3005–3014 (2000)
16. Hinton, G.E.: Training products of experts by minimizing contrastive divergence. Neural Computation 14(8), 1771–1800 (2002)
17. Kawamoto, R., Nazir, A., Kameyama, A., Ichinomiya, T., Yamamoto, K., Tamura, S., Yamamoto, M., Hayamizu, S., Kinosada, Y.: Hidden markov model for analyzing time-series health checkup data. In: MedInfo. Studies in Health Technology and Informatics, vol. 192, pp. 491–495. IOS Press (2013)
18. Kim, H., Marwah, M., Arlitt, M.F., Lyon, G., Han, J.: Unsupervised disaggregation of low frequency power measurements. In: SIAM. pp. 747–758. SIAM / Omnipress (2011)
19. Klopfert, F., Wallenborn, G.: Empowering consumers through smart metering. BEUC, The European Consumer Organisation (2011)
20. Kolter, J.Z., Ferreira, J.: A large-scale study on predicting and contextualizing building energy usage. In: AAAI. AAAI Press (2011)
21. Kolter, J.Z., Jaakkola, T.: Approximate inference in additive factorial hmms with application to energy disaggregation. In: AISTATS. JMLR Proceedings, vol. 22, pp. 1472–1482. JMLR.org (2012)
22. Martinez Alvarez, F., Troncoso, A., Riquelme, J., Aguilar Ruiz, J.: Energy time series forecasting based on pattern sequence similarity. Knowledge and Data Engineering, IEEE Transactions on 23(8), 1230–1243 (Aug 2011)
23. McKerracher, C., Torriti, J.: Energy consumption feedback in perspective: integrating australian data to meta-analyses on in-home displays. Energy Efficiency 6(2), 387–405 (2013), http://dx.doi.org/10.1007/s12053-012-9169-3
24. Mirowski, P., Chen, S., Kam Ho, T., Yu, C.N.: Demand forecasting in smart grids. Bell Labs Technical Journal 18(4), 135–158 (2014), http://dx.doi.org/10.1002/bltj.21650

25. Rabiner, L.R.: Readings in speech recognition. In: Waibel, A., Lee, K.F. (eds.) Readings in speech recognition, chap. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, pp. 267–296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1990), http://dl.acm.org/citation.cfm?id=108235.108253
26. Samuel, I.A., Awelewa, A.A., et al.: Medium-term load forecasting of covenant university using the regression analysis methods. Journal of Energy Technologies and Policy 4(4), 10–16 (2014)
27. Shen, W., Babushkin, V., Aung, Z., Woon, W.L.: An ensemble model for day-ahead electricity demand time series forecasting. In: Proceedings of the Fourth International Conference on Future Energy Systems. pp. 51–62. e-Energy '13, ACM, New York, NY, USA (2013), http://doi.acm.org/10.1145/2487166.2487173
28. Shi, S., Weigend, A.: Taking time seriously: Hidden Markov experts applied to financial engineering. In: CIFEr '97: Proc. of the Conf. on Computational Intelligence for Financial Engineering. pp. 244–252. IEEE (1997)
29. Stern, D.I.: The role of energy in economic growth. Annals of the New York Academy of Sciences 1219(1), 26–51 (2011), http://dx.doi.org/10.1111/j.1749-6632.2010.05921.x
30. Taban, G., A. C?rdenas, A.: Data aggregation as a method of protecting privacy in smart grid networks. Power Systems, IEEE Transactions on (March 2012)
31. Taylor, G.W., Hinton, G.E.: Products of hidden markov models: It takes n ¿1 to tango. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. pp. 522–529. UAI '09, AUAI Press, Arlington, Virginia, United States (2009), http://dl.acm.org/citation.cfm?id=1795114.1795175
32. Wijaya, T.K., Eberle, J., Aberer, K.: Symbolic representation of smart meter data. In: Proceedings of the Joint EDBT/ICDT 2013 Workshops. pp. 242–248. EDBT '13, ACM, New York, NY, USA (2013), http://doi.acm.org/10.1145/2457317.2457357
33. Wijaya, T.K., Ganu, T., Chakraborty, D., Aberer, K., Seetharam, D.P.: Consumer segmentation and knowledge extraction from smart meter and survey data. In: SIAM International Conference on Data Mining (SDM14). pp. 226–234. No. EPFL-CONF-196276 (2014), https://github.com/trtritri/consumer-segmentation
34. Zhang, Z., Ye, S.: Long term load forecasting and recommendations for china based on support vector regression. In: Information Management, Innovation Management and Industrial Engineering (ICIII), 2011 International Conference on. vol. 3, pp. 597–602 (Nov 2011)