Unsupervised Energy Disaggregation with Factorial Hidden Markov Models Based on Generalized Backfitting Algorithm

Lanruo Wang[1], Xianjue Luo[2], Wei Zhang[3]
School of Electrical Engineering
Xi'an Jiaotong University
Xi'an 710049, China
[1]wanglanruo@stu.xjtu.edu.cn
[2]luoxj@mail.xjtu.edu.cn
[3]nothingtodo.zw@stu.xjtu.edu.cn

***Abstract*—This paper proposes an approximate inference and parameter estimation in factorial hidden Markov models (FHMMs), a generalization of hidden Markov models (HMMs) in which the state is factored into multiple Markov chains. Earlier research has proved exact inference and parameter estimation can be computationally intractable. The proposed method in this paper considers FHMMs to be an alternative generalization of generalized additive models (GAMs). We use backfitting algorithm to estimate the parameters in FHMMs instead of exact but complex derivation and make the approximate inference more efficient. This method is motivated by the problem of energy disaggregation which is the process of decomposing a whole household's electricity consumption into individual appliances in order to improve energy utility efficiency both in load terminal and electricity supplier. Numerical simulations indicate the effectiveness of the proposed method for energy disaggregation.**

Keywords—Energy disaggregation; Factorial hidden Markov models; Generalized additive models; Smart metering

I. INTRODUCTION

Electricity is in a large proportion of energy consumption, its efficiency has a directly impact on energy shortage and greenhouse gas emissions. In addition to improving power generation efficiency, carrying out energy saving work in user terminal is also a positive response to energy crisis. Researches showed that with proper feedback and behavioral guidance, household emissions could be reduced by 20% directly[1]. Energy disaggregation is the process of decomposing whole families' energy usage into individual appliances usage by a total electric signal (such as power, voltage, current, etc.), in order to help users make reasonable electricity-using plans under the TOU and sorted energy measure policy in smart grid.

Nonintrusive appliance load monitoring(NIALM) was first proposed by Hart in 1990s[2]. In this century, the research of NIALM tends to study more sophisticated algorithm with the consideration of reduction on additional hardware costs. Akbar et al. suggested a FTT spectrum analysis method with a high frequency sampling[3]. Laughman et al. gave a higher harmonic analysis in the aggregate signals[4]. Suzuki et al. introduced a load monitoring technique based on integer programming[5]. However, these studies neglected some important elements such as additional instrumentation costs. Kim *et al.* proposed that a low sampling rates method could use signals from smart meter directly, in which case no additional instrument is required[6]. Although much work has been done on the extension of additional features of aggregated signals, little attention has been paid on algorithm simplification. This paper focuses on a simplified algorithm to disaggregate power load with time-series data, using factorial hidden Markov models (FHMMs) based on firm statistical foundation, which is known as generalized additive models (GAMs).

The rest of the paper is organized as follows. In Section II, we describe our models for FHMMs based on GAMs, especially the modification of Baum-Welch algorithm for parameter learning. In Section III, simulated results of whole-home load disaggregation are present. Finally, conclusions and possible extensions appear in Section IV.

II. FHMMS BASED ON GAMS

A. FHMMs and GAMs

FHMMs is an extension of HMMs for modeling time series data generated by independent Markov dynamic process[7]. In a FHMM, an observation sequence $Y_t : t = 1,2,...,T\}$ is modeled by specifying probabilistic relationships between observations and hidden states of each relaxed coupled hidden Markov chain. For a FHMM, each trust network is independent, but the observation depends on the current states of all layers. Fig.1 shows a FHMM dynamic trust network with 3 underlying Markov chains.

Consider a state variable matrix S_t :

$$S_t = (S_t^{(1)}, S_t^{(2)}, ..., S_t^{(m)}, ..., S_t^{(M)}) \tag{1}$$

Where, each element represents the m*th* Markov chain's state at time t. Each Markov chain state vector can be expressed as:

$$S^{(m)} = (S_1^{(m)}, S_2^{(m)}, ..., S_t^{(m)}, ..., S_T^{(M)}) \tag{2}$$

Where $\mathrm{m} \in \{1 \ldots M\}$ a represents the number of layer.

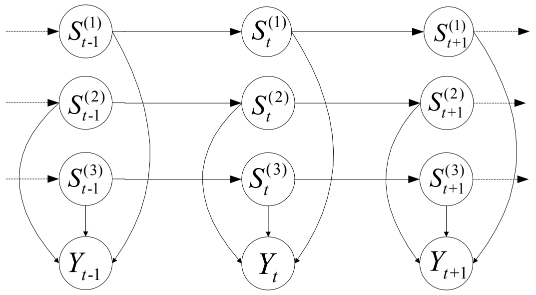


Fig.1.The dynamic trust network of a FHMM with 3 underlying Marko chains

Although FHMM is such a powerful timing analysis model, the traditional EM algorithm used to find parameters' accurate maximum likelihood estimation is usually not tractable.

Fortunately, approximate inference and parameter estimation in FHMMs are very similar to the training process of GAMs. GAMs generates a method for mapping covariate or input variables to response or output variables[8, 9]. GAMs could reflect the statistical dependencies between variables in any setting in which generalized linear models are applicable, such as linear, logistic, or log-linear regression *et al.*. For the mentioned problem above, we could use GAMs to establish relationship between responses and predictors. The general form of GAMs is:

$$g(\mu) = \sum_{m=1}^{M} f_m(S^{(m)}) \tag{3}$$

Where $\mu = E(Y_t \mid S^{(1)}, S^{(2)}, ..., S^{(M)})$ is expectation of Y_t, $\mathrm{g}(\cdot)$ is the connectivity function(its inverse is smooth function). For each predictor $S^{(m)}$, $f_m(\cdot)$ is an arbitrary single variable function generally uses nonparametric methods for fitting, such as cubic spline smoothing function, kernel smoothing function, etc. Parameter estimation of $f_m(\cdot)$ is the main problem in GAMs.

In FHMMs, we can use $f_m(\cdot)$ to determine the additive contribution to observations' expectation of each state variable. By taking the inverse of the function $\mathrm{g}(\cdot)$, the expectations of observations can be obtained by sampling from an appropriate distribution with the given conditional mean.

B. Baum-Welch algorithm for HMMS based on GAMs

Generally, the progress by which FHMMs generates data is similar to the process by which GAMs reflects the relationship between responses and predictor variables. So we consider using the GAMs method for approximate parameter estimation in FHMMs. However, the most significant difference between GAMs and FHMMs is that each stage of learning in GAMs considers the values of covariate variables to be known, but FHMMs consider the hidden state variables are unknown. So the parameter estimation algorithm in GAMs cannot be used directly in FHMMs without any modifications. In the present paper, we use the expected values of the state variables instead of the unknown variables, and we still use Baum-Welch algorithm to compute the expected value of hidden states.

A generalized backfitting algorithm is applied to caculate the additive regression weights to estimate parameters, and this method can be called FHMMs based on GAMs. First a partial residual to modify the m*th* Markov chain at time t is defined:

$$r_t^{(m)} = Z_t - \sum_{i \neq m} f_i(E[S_t^{(i)}]) \tag{4}$$

Where Z_t is a compensation variable of responses, in the real valued models it is considered to be the response variable Y_t, $E[S_t^{(i)}]$ is the expected value of hidden state variable for *ith* chain, and $f_i(E[S_t^{(i)}])$ is the expected contribution of chain *i*. Then this partial residual is used to train the m*th* Markov chain to make the contribution of this chain more approximate to the residual $r_t^{(m)}$. All Markov chains repeat the process for T time steps, and then the training of FHMMs could be accomplished.

Naturally, traditional Baum-Welch algorithm in HMMs should be modified so that the process of backfitting can be integrated in HMMs' parameters estimation. Compensation variable of responses and the weights would affect the training process, so as to achieve the purpose of approaching the residuals. Since all observed variables HMMs are unknown, we suppose that these observed variables to be inferred from Gaussian probability density functions, these functions' parameters can be obtained by statistical learning methods. The revaluated model parameters are given as follows:

First, two variables are defined as follows:

$$\gamma_t^{(m)}(i,l) = P(q_t^{(m)} = i, l_t^{(m)} = l \mid \{r_t^{(m)}\}_{t=1}^{T}, \lambda) \tag{5}$$

$$\xi_t^{(m)}(i,j) = P(q_t^{(m)} = i, q_{t+1}^{(m)} = j, \mid \{r_t^{(m)}\}_{t=1}^{T}, \lambda) \tag{6}$$

$\gamma_t^{(m)}(i,l)$ is a mixed Gaussian probability when the state of m*th* Markov chain is $q_t^{(m)} = i$ and the sample sequence is the l*th* Gaussian mixture component. $\xi_t^{(m)}(i,j)$ is the probability that component *i* is active at time t and component *j* is active at time *t+1* given the residual time series.

According to the maximum likelihood principle, we take the derivative of the Gaussian mixture log-likelihood equation; the process is as same as parameter estimation in traditional HMMs. We describe revaluated formula adding weight and partial residual into the training process as follows, the differences between traditional Baum-Welch and revised formula should be concerned:

$$\pi_i^{(m)} = \frac{P(\{r_t^{(m)}\}_{t=1}^{T}, q_1 = i \mid \lambda)}{P(\{r_t^{(m)}\}_{t=1}^{T} \mid \lambda)} \tag{7}$$

$$a_{i,j}^{(m)} = \frac{\sum_{t=1}^{T-1} \xi_t^{(m)}(i,j)\omega_t}{\sum_{t=1}^{T-1} \gamma_t^{(m)}(i,l)\omega_t} \tag{8}$$

$$c_{i,l}^{(m)} = \frac{\sum_{t=1}^{T} \gamma_t^{(m)}(i,l)\omega_t}{\sum_{t=1}^{T}\sum_{l=1}^{L} \gamma_t^{(m)}(i,l)\omega_t} \tag{9}$$

$$\mu_{i,l}^{(m)} = \frac{\sum_{t=1}^{T} \gamma_t^{(m)}(i,l)\omega_t r_t^{(m)}}{\sum_{t=1}^{T} \gamma_t^{(m)}(i,l)\omega_t} \quad (10)$$

$$\Sigma_{i,l}^{(m)} = \frac{\sum_{t=1}^{T} \gamma_t^{(m)}(i,l)\omega_t (r_t^{(m)} - \mu_{i,l}^{(m)})(r_t^{(m)} - \mu_{i,l}^{(m)})^T}{\sum_{t=1}^{T} \gamma_t^{(m)}(i,l)\omega_t} \quad (11)$$

L in (9) denotes the numbers of Gaussian components in Gaussian mixture.

Thus we can derive the probability density function of observations:

$$b_i^{(m)}(o) = \sum_{l=1}^{L} c_{i,l}^{(m)} b_{i,l}(o) \quad (12)$$

Where $c_{i,l}^{(m)}$ is output mixing ratio of $b_{i,l}(o)$, and $\sum_{l=1}^{L} c_{i,l}^{(m)} = 1$. The density function of every $b_{i,l}(o)$ is:

$$b_{i,j}(o) = (2\pi)^{-\frac{D}{2}} \left|\Sigma_{i,j}^{(m)}\right|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(o-\mu_{i,l}^{(m)})^T \Sigma_{i,j}^{(m)} (o-\mu_{i,l}^{(m)})\right) \quad (13)$$

The pseudo-code in Fig.2 outlines training procedure of FHMMs based on GAMs:

1. For 1 ~ M Markov chains, construct compensation variable of responses $\{Z_t\}_1^T$ and compute weights $\{\omega_t\}_1^T$;
2. Compute the partial residual $\{R_t^{(m)}\}_1^T$;
3. Use weights $\{\omega_t\}_1^T$ and residual $\{R_t^{(m)}\}_1^T$ as parameters, then train each Markov chain using Baum-Welch algorithm until convergence reaches the set number of iterations;
4. Use means of each Markov chain to calculate of the means and variances, initial probability, transition probabilities of FHMM, check whether the convergence criteria is set or the number of iterations is achieved, otherwise returns 1.

Fig.2. The FHMM algorithm based on GAM

III. Proposed Model for Energy Disaggregation

As mentioned above, our ultimate goal is to disaggregate whole home energy into appliances behavior. In order to test the effectiveness of the algorithm, we have simulated electrical behavior of two families and describe the summary of each household appliance in TABLE1. To make our simulation more closer to actual situations, the REDD data sets in [10] are referenced. The simulated appliances' on-durations are sampled from gamma distributions, and off-durations are from bimodal gauss distributions. Here we considered two-state appliances. We first simulate switching behavior and on/off durations of each household's appliances, then aggregate the data from multiple individual appliances. We use the aggregated power to test the ability of the method mentioned above, it is important to note that each appliances measurements are unnecessary to obtain in the disaggregating, we just use them to help us find out the ground truth and check our algorithm, the algorithm remains nonintrusive.

TABLE I. Appliances of 3 families

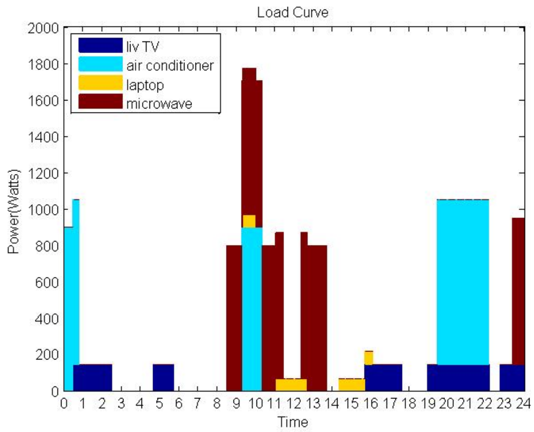
a. Appliances of Fam.1

Number	***Name of appliance***	***Power/W***
1	Television-1	150
2	Air conditioner	900
3	Laptop	70
4	Microwave	800

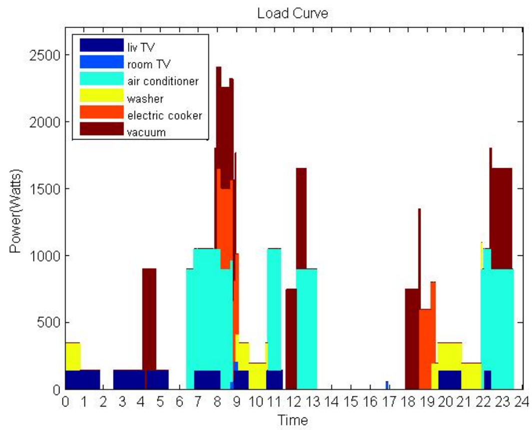
b. Appliances of Fam.2

Number	***Name of appliance***	***Power/W***
1	Television-1	150
2	Television-2	60
3	Air conditioner	900
4	Washing-machine	200
5	Electric cooker	600
6	Vacuum cleaner	750

Fig.3 shows the simulated one-day appliances activities, Fig.4 shows the corresponding predicted breakdown situations using the method proposed above. Although the judgment of some appliances' start and end time was not entirely accurate, the basic electrical switch recognition had already been achieved.

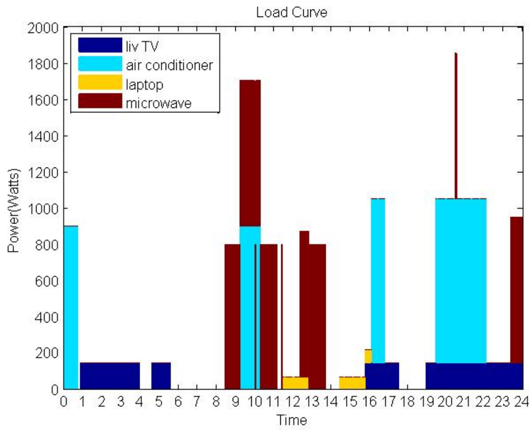


a. One-day appliances activities of Fam.1

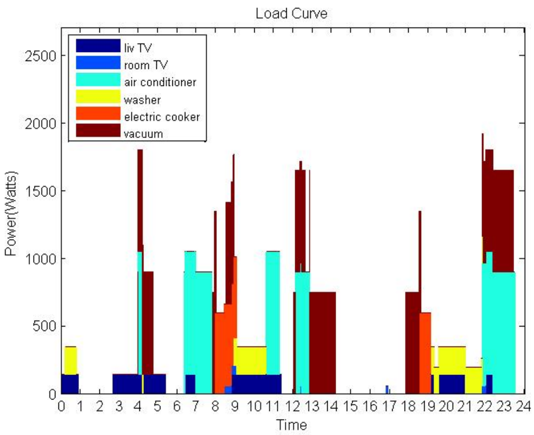


b. One-day appliances activities of Fam.2

Fig.3. Simulated Appliance activities of 3 families



a. Disaggregated One-day appliances activities of Fam.1



b. Disaggregated One-day appliances activities of Fam.2

Fig.4. Disaggregated Appliance activities of 3 families

We use an evaluation index that is always used in the filed of information retrieval evaluation, F-Measure, to evaluate the accuracy of the decomposition method. Generally, F-Measure is calculated as:

$$F=\frac{(a^2+1)P\times R}{a^2(P+R)} \tag{14}$$

Where, $P=\frac{tp}{tp+fp}$ represents Precision, and $R=\frac{tp}{tp+fn}$ represents Recall. We use the following statistic in large-scale data collection to calculate P and R:

TABLE II. STATISTIC NEEDED IN F-MEASURE

	Relevant	Nonrelevant
Retrieved	true positive (tp)	false positive (fp)
Not retrieved	false negative (fn)	true negative (fn)

According to the original traces of power use for each appliance, we compute F-Measures of the decomposed data, as a comparison, we also used the exact EM algorithm to train FHMMs. As can be seen from TABLE III, the performance of FHMMs based on GAMs is comparable to exact EM algorithm, FHMM-EM performs better is unsurprising because the algorithm performs exact inference and maximum likelihood parameter estimation. The results are highly compatible with the results of exact EM.

TABLE III. PERFORMANCE FOR FHMM-GAM AND EXACT-EM FOR ONE-DAY AGGREGATED DATA FROM FAMILY 1 AND FAMILY 2

a. F-MEASURE OF FHMM-GAM AND EXACT-EM FOR FAM.1

Name of appliance	*FHMM-GAM*	*FHMM-EM*
1 Television-1	0.6679	0.6035
2 Air conditioner	0.5187	0.8104
3 Laptop	0.5898	0.8306
4 Microwave	0.6429	0.8389
Total	0.6879	0.7708

b. F-MEASURE OF FHMM-GAM AND EXACT-EM FOR FAM.2

Name of appliance	*FHMM-GAM*	*FHMM-EM*
1 Television-1	0.6062	0.5285
2 Television-2	0.5543	0.9757
3 Air conditioner	0.5069	0.5437
4 Washing-machine	0.6118	0.7854
5 Electric cooker	0.7312	0.5845
6 Vacuum cleaner	0.6118	0.6938
Total	0.5904	0.6654

IV. CONCLUSION

This paper presents an alternative approach to NIALM capable of disaggregating a whole-home total load into individual appliances behaviors, using an approximate method for parameter estimation and inference of FHMMs based on GAMs. We describe a revaluated formula adding weight and partial residual into the training process of traditional Baum-Welch algorithm and express the revised formula. Through evaluation using simulated aggregate and individual data for two households, we have proved that the proposed approach can accomplish the task of load decomposition, and could achieve a compatible performance of exact EM algorithm under the same conditions. As an extension of this work, future work will look at models with multiple-state appliances, and we will also try to estimate more devices in the home, and consider adding extra features such environmental and regional factors which would affect people's consumption habits significantly.

REFERENCES

[1] T. Dietz, G. T. Gardner, J. Gilligan, P. C. Stern, and M. P. Vandenbergh, "Household actions can provide a behavioral wedge to rapidly reduce US carbon emissions," *Proceedings of the National Academy of Sciences,* vol. 106, pp. 18452-18456, 2009.

[2] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the Ieee,* vol. 80, pp. 1870-1891, 1992.

[3] M. Akbar and D. Z. A. Khan, "Modified Nonintrusive Appliance Load Monitoring For Nonlinear Devices," in *Multitopic Conference, 2007. INMIC 2007. IEEE International*, 2007, pp. 1-5.

[4] C. Laughman*, et al.*, "Power signature analysis," *Power and Energy Magazine, IEEE,* vol. 1, pp. 56-63, 2003.

[5] K. Suzuki, S. Inagaki, T. Suzuki, H. Nakamura, and K. Ito, "Nonintrusive appliance load monitoring based on integer programming," in *SICE Annual Conference, 2008*, 2008, pp. 2742-2747.

[6] H. S. KIM, "Unsupervised disaggregation of low frequency power measurements," University of Illinois, 2012.

[7] Z. Ghahramani and M. I. Jordan, "Factorial Hidden Markov Models," *Machine Learning,* vol. 29, pp. 245-275, 1997.

[8] R. A. Jacobs, W. Jiang, and M. A. Tanner, "Factorial hidden markov models and the generalized backfitting algorithm," *Neural Computation,* vol. 14, pp. 2415-2437, 2002.

[9] T. Hastie and R. Tibshirani, *Generalized additive models* vol. 43: Chapman & Hall/CRC, 1990.

[10] J. Z. Kolter and M. J. Johnson, "REDD: A Public Data Set for Energy Disaggregation Research," 2011.