Deep Online Hierarchical Dynamic Unsupervised Learning for Pattern Mining from Utility Usage Data

Saad Mohamad[a,*], Abdelhamid Bouchachia[a]

[a]*Department of Computing, Bournemouth University, Poole, UK*

Abstract

While most non-intrusive load monitoring (NILM) work has focused on supervised algorithms, unsupervised approaches can be more interesting and practical. Specifically, they do not require labelled training data to be acquired from the individual appliances and can be deployed to operate on the measured aggregate data directly. We propose a fully unsupervised novel NILM framework based on Dynamic Bayesian hierarchical mixture model and Deep Belief network (DBN). The deep network learns, in unsupervised fashion, low-level generic appliance-specific features from the raw signals of the house utilities usage, then the hierarchical Bayesian model learns high-level features representing the consumption patterns of the residents captured by the correlations among the low-level features. The temporal ordering of the high-level features is captured by the Dynamic Bayesian Model. Using this architecture, we overcome the computational complexity that would occur if temporal modelling was directly applied to the raw data or even to the constructed features. The computational efficiency is crucial as our application involves massive data from different utilities usage. Moreover, we develop a novel online inference algorithm to cope with this big data. Finally, we propose different evaluation methods to analyse the results which show that our algorithm finds useful patterns.

Keywords: Non-intrusive load monitoring, Bayesian modelling, online

[*]Saad Mohamad is the corresponding author

Email addresses: smohamad@bournemouth.ac.uk (Saad Mohamad), abouchachia@bournemouth.ac.uk (Abdelhamid Bouchachia)

learning, Human activity recognition.

1. Introduction

The monitoring of human behaviour is highly relevant to many real-word domains such as safety, security, health and energy management. Research on human activity recognition (HAR) has been the key ingredient to extract pattern of human behaviour. HAR research can be categorised into three main classes: sensor-based [1], vision-based [2] and radio-based [3]. A common feature of these methods is that they all require equipping the living environment with embedded devices (sensors). On the other hand, non-intrusive load monitoring (NILM) requires only single meter per house or a building that measures aggregated signals at the entry point of the meter. Various techniques can then be used to disaggregate per-load power consumption from this composite signal providing energy consumption data at an appliance level granularity. In this sense, NILM focuses not on extracting general human behaviour patterns but rather on identifying the appliances in use. This, however, can provide insight into the energy consumption behaviour of the residents and therefore can express users life style in their household. The idea of abandoning the high costs and management/maintenance induced by various sensors entailed by traditional HAR studies makes NILM an attractive approach to exploit in general pattern recognition problems. On the other hand, taking the human behaviour into account can leverage the performance of NILM; thus, providing finer understanding of the resident's energy consumption behaviour. In this paper, we do not distinguish between patterns and appliances recognition. The main goal of our approach is to encode the regularities in a massive amount of energy consumption data into a low-dimensional representation. This is only possible by the fact that human behaves orderly following certain patterns. We are also lucky to have an extra large amount of real-world data which makes this approach more viable.

Since the earliest work on NILM [4], most NILM work has been based on signal processing and engineering approaches [5, 6]. This can explain the fact that even with the economical attractive tools that NILM can provide for pattern recognition and HAR communities, it has not been widely exploited. Most of existing machine learning approaches to NILM adopt supervised algorithms [4, 7, 8, 9, 10, 11, 12, 13]. Such algorithms could damage the attractiveness of NILM as they require individual appliance data for training, prior to the system deployment. Hence, there is a need to install one energy meter per appliance to record appliance-specific energy consumption. This incurs extra costs and a complex installation of sensors on every device of interest. In contrast, unsupervised algorithms can be deployed to operate directly from the measured aggregate data with no need for annotation. Hence, unsupervised algorithms are clearly more suitable for NILM. To the best of our knowledge, all existing unsupervised approaches to NILM [14] concentrate on disaggregating the whole house signal into its composing (appliances') ones. In contrast, our approach, as mentioned earlier, does not focus on identifying per-appliance signal. We instead propose a novel approach that seeks to extract human behaviour patterns from home utility usage data. These patterns could be exploited for HAR as well as energy efficiency analysis.

The proposed approach is a three-module architecture composed of a DBN, a hierarchical Bayesian mixture model based on Latent Dirichlet Allocation (LDA) and a Dynamic Bayesian Network model based on Bayesian Hidden Markov Model (HMM). Hence, we call it DBN-LDA-HMM. It draws inspiration from the work in [15, 16]. Authors in [15] plug a hierarchical Dirichlet process (HDP) prior on top of a Deep Boltzmann Machine network which allows learning multiple layers of abstractions. The low-level abstraction represents generic domain-specific features that are hierarchically clustered and shared to yield high-level abstraction representing patterns. However, this model does not consider the temporal ordering of the high-level representations (patterns). On the other hand, the work in [16] proposed an LDA-HMM hybrid model to perform action recognition. The model was motivated by the success and the efficiency of the bag-of-words approach, adopted by topic modelling, in solving general high-level problems. The temporal ordering power of HMM is harnessed to correlate the activity at high level. The paper uses collapsed Gibbs sampler for approximate inference and learning. We employ an unsupervised version of this model similar to the one introduced in [17]. However, we propose a stochastic variational inference (SVI) [18] algorithm that allows scalable inference to cope with the massive amount of energy consumption data (around 80 TB). We also employ the DBN to construct appliance-specific features which are used by the hierarchical Bayesian mixture model to construct components (topic)-specific features. Mixtures of these components form the residents' energy consumption patterns. The dynamic modelling part exploits the temporal regularity in the human behaviour leading to better performance and allowing forecasting.

Recently, the field of deep learning (DL) has made a huge impact and achieved remarkable results in computer vision, natural language processing, and speech recognition. Yet it has not been exploited in the field of NILM. DL provides an effective tool for extracting multiple layers of distributed features representations from high-dimensional data. Each layer of the deep architecture performs a non-linear transformation of the outputs of the previous layer. Thus, through DL, the data is represented in the form of a hierarchy of features, from low to high level [19, 20]. Instead of relying on heuristic hand-crafted features, DL learns to extract features that allow for more discriminative power. Supported by the sheer size of the available data and its high sampling rate (205 KHZ) which results in a very high-dimensional data, we are the first to use unsupervised DL model in NILM. In contrast to existing electrical engineering and signal processing approaches adopted in NILM, ours relies fully on the data to construct informative features.

In this paper, we pre-train a DBN [21] to learn generic features from unlabelled raw electrical signal with 1 second granularity. The extracted features are fed to the LDA-like part of the the model with 30 minutes granularity. Although, the bag-of-words assumption adopted here is a major simplification, it break down unnecessary low-level hard-to-model complexity leading to computationally efficient inference with no much loss as shown in LDA [22]. Finally, an easy-to-model dynamic is done by the HMM-like part of the model.

In this work, we demonstrate that this approach can capture significant statistical structure in a specified window of data over a period of time. This structure provides understanding of regular patterns in the human behaviour that can be harnessed to provide various services including services to improve energy efficiency. For example, understanding the usage and energy consumption patterns could be used to predict the power demand (load forecasting), to apply management policies and to avoid overloading the energy network. Moreover, providing consumers with information about their consumption behaviour and making them aware of abnormal consumption patterns compared to others can influence their behaviour to moderate energy consumption [23].

As already mentioned, this algorithm is designed to be trained over a very huge amount of data resulting from the high sampling rate around 205 kHz of the electricity signal which gives us an advantage compared to the data used in other research studies except for [24, 25, 26]. Besides the advantage the data size offers, apart from [27, 28] whose sampling rate is very low, our data is the only one including water and gas usage data. Moreover, measurements provided by additional sensors are also exploited to refine the performance of the pattern recognition algorithm. More details on the data can be found in Sec. 4. The diversity of the data is another motivation for adopting a pattern recognition approach rather than traditional disaggregation approach. In a nutshell, we propose an original NILM method with three characteristics:

- Scalability to deal with massive high-dimensional data. The proposed three-module architecture model can learn from high-dimensional raw data and uses online learning to cope with the massive size of the data.

- Unsupervised learning. No labelling of appliance data is required. The method bridges the gap between pattern recognition and NILM allowing for sensors-free pattern recognition

- Learning from massive heterogeneous data. The data used in this study comes from multi-utility usage.

The rest of the paper is organised as follows. Section 2 presents the related work. Section 3 presents the proposed approach. Section 4 describes the data and discusses the obtained results. Finally, Sec. 5 concludes the paper and hints to future work.

2. Related Work

We divide the related work into two parts: (i) machine learning approaches to NILM and (ii) NILM data used in the literature.

As we have discussed in the introduction, most of existing NILM studies are not based on machine learning algorithms and most of machine learning NILM algorithms are supervised ones [4, 7, 8, 9, 10, 11, 12, 13, 29, 30, 31, 32]. Such algorithms requires training on labelled data which is expensive and laborious to obtain. In fact, the practicality of NILM is stemmed from the fact that it comes with almost no setup cost. Recently, researchers have started exploring unsupervised machine learning algorithms to NILM. These methods have mainly focused on performing energy disaggregation to discern appliances from the aggregated load data directly without performing any sort of event detection. The most prominent of these methods are based on Dynamic Bayesian Network models, in particular different variants of Hidden Markov Model (HMM) [33, 34, 35].

Authors in [33] proposes to use Factorial Hidden Markov Model (FHMM) and three of its variants: Factorial Hidden Semi-Markov Model (FHSMM), Conditional FHMM (CFHMM) and Conditional FHSMM (CFHSMM) to achieve energy disaggregation. The main idea is that the dynamics of the state occupancy of each appliance evolves independently and the observed aggregated signal is some joint function of all the appliances states. To better model the state occupancy duration, that is modelled with a geometric distribution by FHMM, authors propose to use FHSMM which allows modelling the durations of the appliances states with gamma distribution. Authors also propose CFHMM to incorporate additional features, such as time of day, other sensor measurements, and dependency between appliances. To harness the advantages of FHSMM and CFHMM, authors propose a combination of the two models resulting in CFHSMM. In that work, the electricity signal was sampled at low frequency which is in contrast to our work.

Similar approach was taken in [34] where Additive Factorial Hidden Markov Model (AFHMM) was used to separate appliances from the aggregated load data. The main motivation and contribution of this approach is that it addresses the local optima problems that existing approximate inference techniques [33] are highly susceptible to experience. The idea is to exploit the additive structure of AFHMM to develop a convex formulation of approximate inference that is more computationally efficient and has no issues of local optima. Although, this approach was applied on relatively high frequency electricity data [26], the data scale is not close to ours. Hierarchical Dirichlet Process Hidden Semi-Markov Model is used in [35] to incorporate duration distributions (Semi Markov) and allows to infer the number of states from the data (Hierarchical Dirichlet Process). On the contrary, the AFHMM algorithm in [34] requires the number of appliances (states) to be set a-priori. The work by [36] uses iterative fuzzy c-means to determine the number of hidden states.

The common feature of the approaches discussed so far is that the considered data sets are collected only from the electricity signals. In contrast, our data involves different utilities namely electricity, water and gas data as well some sensors measurements that provide contextual features. To the best of our knowledge, the only data that considers water and gas usage data is [27, 28]. However, the sampling rate of this data is very low compared to ours. Authors in [37] exploit the correlation between appliances and side information, in particular temperature, in a convex optimisation problem for energy disaggregation. This algorithm is applied on low sampling rate electricity data with contextual supervision in the form of temperature information.

This work is a continuation of our previous work [38, 39]. In [38], online Gaussian Latent Dirichlet Allocation (GLDA) is proposed to extract global components that summarise the energy signal. These components provide a rep-

Table. 1 Relevant work on NILM

Algorithms	Unsupervised	Scalability	heterogeneous data	temporal dependency
[12, 29, 30]		✓		
[33, 34, 35]	✓			✓
[27, 28]	✓		✓	
[37]			✓	✓
[38, 39]	✓	✓	✓	
ours	✓	✓	✓	✓

resentation of the consumption patterns. The algorithm is applied on the same data-set as in this paper. However, in contrast to [38], DBN-LDA-HMM employs deep learning to construct features rather than engineering them using signal processing technique. Similarly, the work by [39] also employs deep learning for features extraction. However, DBN-LDA-HMM also considers temporal dependency. To wrap up this section, four features distinguish our approach from existing ones. It bridges the gap between pattern recognition and NILM making it beneficial for a variety of different applications. Driven by massive amount of data, our method is computationally efficient and scalable, unlike state-of-the-art probabilistic methods that posit detailed low-level temporal relationships and involve complex inference steps. The approach is fully data-driven where DL is used to learn the features unlike existing features engineering approach. The available data has a high sampling rate electricity data allowing learning more informative features. It also includes data from other utility usage and additional sensors measurements. Thus, our work also covers the research aspect of NILM concerned with the acquisition of data, prepossessing steps and evaluation of NILM algorithms.

3. Proposed Approach

Learning human behaviour from NILM is very challenging. The data is highly unstructured with sequential dependency. Furthermore, labelling such sequential data is expensive. In this work, we aim at understanding the human behaviour using NILM data. To capture such high-level pattern from highly unstructured unlabelled sequential data, highly informative big data is needed. We collect massive heterogeneous data from multiple utilities usages with high sampling rate.

In this section, we present our proposed novel approach that is capable of learning from our collected massive sequential highly unstructured NILM data. Figure 1 shows the mapping from input raw data via the proposed three-module architecture DBN-LDA-HMM to the extract human consumption behaviour. At the bottom, DBN reduces the high dimensional input and extract low-level structured generic appliances-specific features from the raw signals of the house utilities usage. LDA infers high-level features representing the residents' consumption patterns captured by the correlations among the low-level features. It uses the bag-of-words assumption to allow computationally efficient inference. The temporal ordering of the high-level features is captured by HMM. The multi-layers dimension reduction achieved by this architecture results in tractable computational complexity in contrast to that occurring if HMM was directly applied to the raw data or even the constructed features. Note that although we focus on NILM data, DBN-LDA-HMM is generic and can be used to model any problem involving highly unstructured high dimensional sequential data.

In order to learn from our massive data, we propose a scalable novel inference algorithm. In this section, we focus on the Dynamic Bayesian hierarchical mixture model (LDA-HMM) and its novel inference algorithm. Details about the DL part of the model (Deep Belief Network) can be found in [21] and App. 6. As discussed earlier, similar model is proposed by [17] where inference is done using MCMC sampling method. Conversely, we develop a variational inference (VI) method.

VI has become widely used as a deterministic alternative approach to MCMC sampling. In general, VI tends to be faster than MCMC which makes it more suitable for our large scale problems. VI turns the inference problem to an optimisation problem by positing a simpler family of distributions and finding the member of the family that is closest to the true posterior distribution [40].

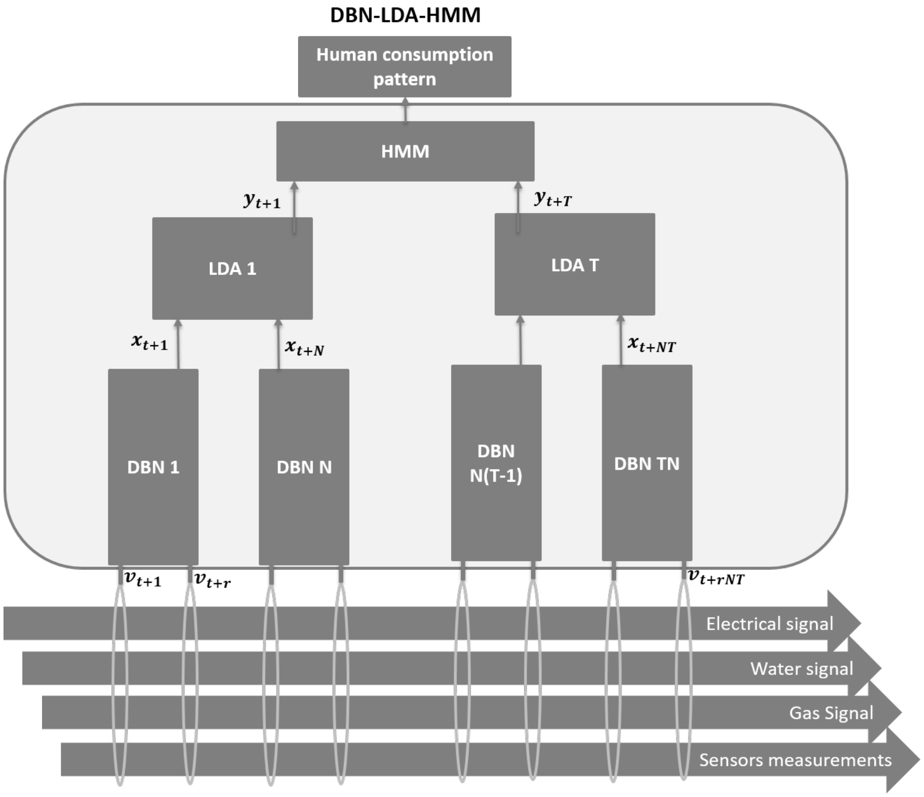


Figure. 1 Model architecture

Hence, the inference task boils down to an optimisation problem of a non-convex objective function. This allows us to bring sophisticated tools from optimisation literature to tackle the performance problems.

Recently, stochastic optimisation has been applied to VI in order to cope with massive data [18]. While VI requires repeatedly iterating over the whole data set before updating the variational parameters (parameters of the variational objective), stochastic variational inference (SVI) updates the parameters every time a single data example is processed. Therefore, by the end of one pass through the dataset, the parameters will have been updated multiple times. Hence, the model parameters converge faster, while using less computational resources. The idea of SVI is to move the variational parameters at each iteration in the direction of a noisy estimate of the variational objective's natural gradient based on a couple of examples [18]. Following these stochastic gradients with certain conditions on the (decreasing) learning rate schedule, SVI provably converges to a local optimum [41].

It can be easily shown that LDA-HMM is a member of the family of graphical models proposed by [18] where observations, global hidden variables, local hidden variables, and fixed parameters are involved. Hence, SVI for LDA-HMM can be derived following similar but more complicated steps as LDA in [18] and HMM in [42]. However, for simplification, we develop tailored SVI to LDA-HMM. In the following, we present the graphical model, its distributions and the proposed SVI algorithm.

As we have mentioned, LDA-HMM is a member of the family of models presented in [18]. The global hidden variables include appliance-related (low-level) variables, patterns-related (high-level) variables and dynamic-related variables. The local hidden variables include HMM "state" selection variables and LDA "topic" selection variables. The state variables are distributed according to Multinomial distribution governed by the global dynamic parameters. They select the patterns generating the topic selection variables which are also Multinomial distribution. Note that the observations are the discrete output of the DL algorithm. The graphical model is shown in Fig. 2. In the following, we list LDA-HMM's variables distribution which also satisfies the conjugacy assumptions of the family of models presented in [18]:

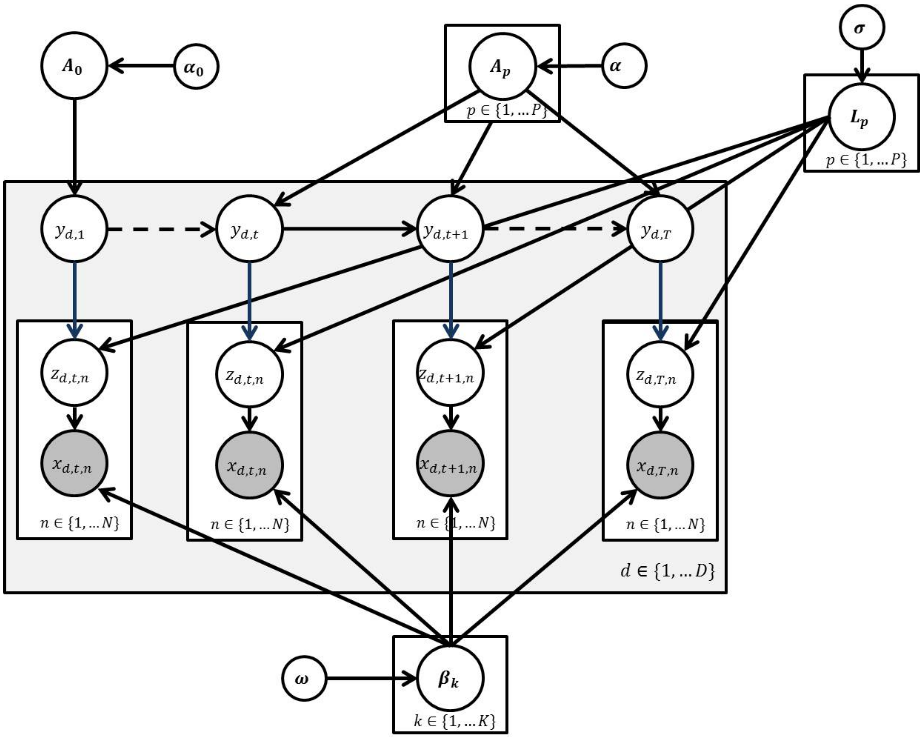


Figure. 2 Graphical Model

$$
\begin{gathered}
\boldsymbol{A_0} | \boldsymbol{\alpha_0} \sim Dir(\boldsymbol{\alpha_0}) \\
\boldsymbol{A_p} | \boldsymbol{\alpha} \sim Dir(\boldsymbol{\alpha}) \\
\boldsymbol{L_p} | \boldsymbol{\sigma} \sim Dir(\boldsymbol{\sigma}) \\
\boldsymbol{\beta_k} | \boldsymbol{\omega} \sim Dir(\boldsymbol{\omega}) \\
y_{d,1} | \boldsymbol{A_0} \sim Mult(\boldsymbol{A_0}) \\
y_{d,t} | y_{d,t-1}, \{\boldsymbol{A_p}\}_{p=1}^{P} \sim Mult(\boldsymbol{A}_{y_{d,t-1}}) \\
z_{d,t,n} | y_{d,t}, \{\boldsymbol{L_p}\}_{p=1}^{P} \sim Mult(\boldsymbol{L}_{y_{d,t}}) \\
x_{d,t,n} | z_{d,t,n}, \{\boldsymbol{\beta_k}\}_{p=1}^{K} \sim Mult(\boldsymbol{\beta}_{z_{d,t,n}})
\end{gathered}
\tag{1}
$$

Our ultimate purpose is to compute the posterior distribution over the hidden variables or some of them. By doing so, we can get insight into the energy consumption behaviour and lifestyle of the residents. However, it can be clearly seen that computing such posterior is intractable and approximation is needed. As we have already mentioned VI turns the inference problem to an optimisation problem by positing a simpler family of distributions, called variational distribution, and minimising the Kullback-Leibler divergence from the actual posterior distribution. This is equivalent to maximising the evidence lower bound (ELBO); a lower bound on the logarithm of the marginal probability of the observations $\log p(\boldsymbol{x})$:

$$\begin{aligned}\log p(\boldsymbol{x}) = \log \int_{\boldsymbol{\beta}} \int_{\boldsymbol{A}} \int_{\boldsymbol{L}} \int_{\boldsymbol{A}_0} \sum_{\boldsymbol{y}} \sum_{\boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}, \boldsymbol{A_0}, \boldsymbol{L}, \boldsymbol{A}, \boldsymbol{\beta}) d\boldsymbol{A_0} d\boldsymbol{L} d\boldsymbol{A} d\boldsymbol{\beta} \\ = \log \int_{\boldsymbol{\beta}} \int_{\boldsymbol{A}} \int_{\boldsymbol{L}} \int_{\boldsymbol{A}_0} \sum_{\boldsymbol{y}} \sum_{\boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}, \boldsymbol{A_0}, \boldsymbol{L}, \boldsymbol{A}, \boldsymbol{\beta}) \\ \frac{q(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}, \boldsymbol{A_0}, \boldsymbol{L}, \boldsymbol{A}, \boldsymbol{\beta})}{q(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}, \boldsymbol{A_0}, \boldsymbol{L}, \boldsymbol{A}, \boldsymbol{\beta})} d\boldsymbol{A_0} d\boldsymbol{L} d\boldsymbol{A} d\boldsymbol{\beta} \qquad (2)\end{aligned}$$

Using Jensens inequality and the concavity of the logarithm on Eq. (2), we can obtain ELBO:

$$\begin{aligned}\log p(\boldsymbol{x}) \geq & E_q[\log p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}, \boldsymbol{A_0}, \boldsymbol{L}, \boldsymbol{A}, \boldsymbol{\beta})] - E_q[\log q(\boldsymbol{z}, \boldsymbol{y}, \boldsymbol{A_0}, \boldsymbol{L}, \boldsymbol{A}, \boldsymbol{\beta})] \\ = & \mathcal{L}(q) \qquad (3)\end{aligned}$$

The mean-field variational family is the commonly used and simplest approximation where each hidden variable is independent and governed by its own parameter. We propose a partial mean-field variational distributions by retaining the dynamic structure of the HMM-part of the model because inference for those variables is tractable using the well-known Forward-backward algorithm. Equation (4) shows the proposed mean-field variational distributions:

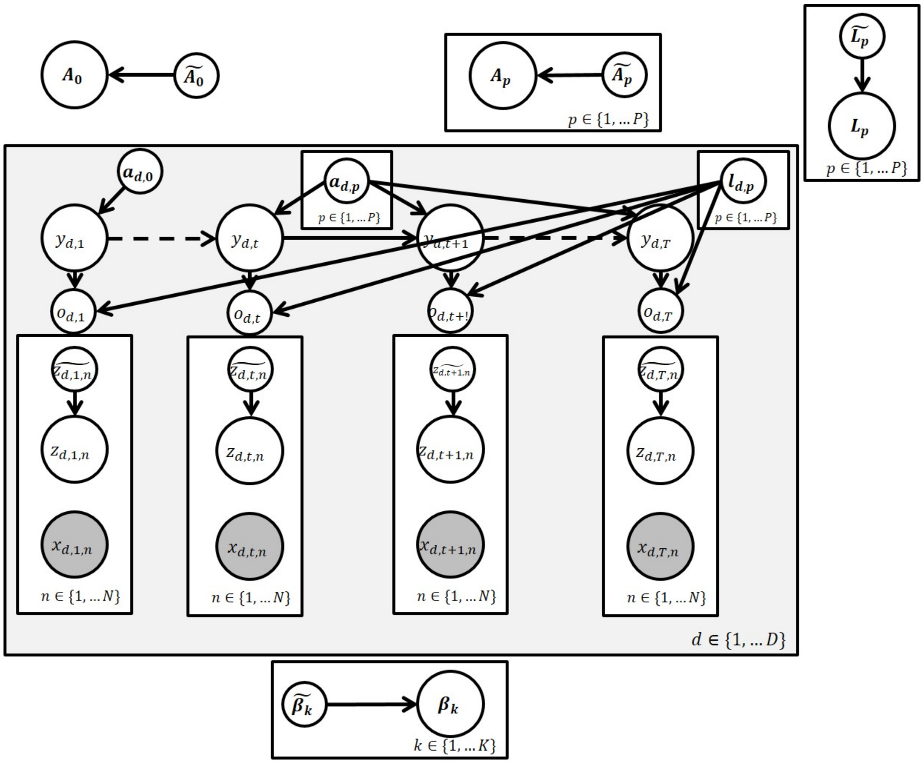


Figure. 3 Graphical Model of the variational approximation

$$q(\boldsymbol{z}, \boldsymbol{y}, \boldsymbol{A_0}, \boldsymbol{L}, \boldsymbol{A}, \boldsymbol{\beta}) = q(\boldsymbol{z})p(\boldsymbol{y})p(\boldsymbol{A_0})p(\boldsymbol{L})p(\boldsymbol{A})p(\boldsymbol{\beta}) \tag{4}$$

where

$$\begin{aligned}
\boldsymbol{A}_{\boldsymbol{0}}|\tilde{\boldsymbol{A}}_{\boldsymbol{0}} &\sim Dir(\tilde{\boldsymbol{A}}_{\boldsymbol{0}})\\
\boldsymbol{A}_{\boldsymbol{p}}|\tilde{\boldsymbol{A}}_{\boldsymbol{p}} &\sim Dir(\tilde{\boldsymbol{A}}_{\boldsymbol{p}})\\
\boldsymbol{L}_{\boldsymbol{p}}|\tilde{\boldsymbol{L}_{\boldsymbol{p}}} &\sim Dir(\tilde{\boldsymbol{L}_{\boldsymbol{p}}})\\
\boldsymbol{\beta}_{\boldsymbol{k}}|\tilde{\boldsymbol{\beta}}_{\boldsymbol{k}} &\sim Dir(\tilde{\boldsymbol{\beta}}_{\boldsymbol{k}})\\
z_{d,t,n}|\tilde{\boldsymbol{z}}_{\boldsymbol{d,t,n}} &\sim Mult(\tilde{\boldsymbol{z}}_{\boldsymbol{d,t,n}})\\
y_{d,1}|\tilde{\boldsymbol{a}}_{\boldsymbol{d,0}} &\sim Mult(\tilde{\boldsymbol{a}}_{\boldsymbol{d,0}})\\
y_{d,t}|y_{d,t-1},\tilde{\boldsymbol{l}}_{\boldsymbol{d,t}},\{\tilde{\boldsymbol{a}}_{\boldsymbol{d,p}}\}_{p=1}^{P} &\sim Mult(diag(\tilde{\boldsymbol{l}}_{\boldsymbol{d,t}})\tilde{\boldsymbol{a}}_{\boldsymbol{d,y_{d,t-1}}})
\end{aligned} \tag{5}$$

where $p(o_{d,t}|y_{d,t},\boldsymbol{l}_{\boldsymbol{d,t}}) = \tilde{l}_{d,t,y_{d,t}}$. Figure 3 shows the relationships between the variables of the variational distributions. Following similar steps as in [18], our goal is to optimise ELBO with respect to the variational parameters. In traditional mean-field variational inference, ELBO is optimised with coordinate ascent, where each variational parameter is iteratively optimise, holding the other parameters fixed. Since LDA-HMM is a member of the family presented in [18], deriving SVI from VI is straightforward as the case in [18]. We first derive the coordinate update for the global parameters $\{\tilde{\boldsymbol{\beta}},\tilde{\boldsymbol{A}},\tilde{\boldsymbol{A}}_{\boldsymbol{0}},\tilde{\boldsymbol{L}}\}$ of the variational distribution, then the local ones $\{\tilde{\boldsymbol{z}},\boldsymbol{a}_{\boldsymbol{0}},\boldsymbol{a},\tilde{\boldsymbol{l}}\}$.

3.1. Global Parameters

As a function of the appliance-related variational parameters $\tilde{\boldsymbol{\beta}}$, we can rewrite the objective as:

$$\mathcal{L}(\tilde{\boldsymbol{\beta}}) = E_q[\log p(\boldsymbol{\beta}|\boldsymbol{x},\boldsymbol{z},\boldsymbol{\omega})] - E_q[\log q(\boldsymbol{\beta}|\tilde{\boldsymbol{\beta}})] + const \tag{6}$$

$$E_q[\log p(\boldsymbol{\beta}|\boldsymbol{x},\boldsymbol{z},\boldsymbol{\omega})] = E_q[\log p(\boldsymbol{x}|\boldsymbol{\beta},\boldsymbol{z})] + E_q[\log p(\boldsymbol{\beta}|\boldsymbol{\omega})] + const \tag{7}$$

$$E_q[\log p(\boldsymbol{x}|\boldsymbol{\beta}, \boldsymbol{z})] = \sum_{d=1}^{D}\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{k=1}^{K} E_q[\log \beta_{k,x_{d,t,i}}]q(z_{d,t,i} = k|\boldsymbol{\tilde{z}_{d,t,n}}) \tag{8}$$

$$E[\log p(\boldsymbol{\beta_k}|\boldsymbol{\omega})] = \sum_{i=1}^{V}(\omega_i - 1)E_q[\log \beta_{k,i}] \tag{9}$$

where V is the number of DL outcomes.

$$E_q[\log q(\boldsymbol{\beta_k}|\boldsymbol{\tilde{\beta}_k})] = \log \Gamma(\sum_{i}^{V} \tilde{\beta}_{k,i}) - \sum_{i}^{V} \log \Gamma(\tilde{\beta}_{k,i}) + \sum_{i=1}^{V}(\tilde{\beta}_{k,i} - 1)E_q[\log \beta_{k,i}] \tag{10}$$

By taking the derivative of the objective function with respect to $\tilde{\beta}_{k,v}$ and set it to zero, we obtain the following:

$$\begin{aligned} &\frac{d\mathcal{L}(\boldsymbol{\tilde{\beta}})}{d\tilde{\beta}_{k,v}} = 0 \implies \sum_{d=1}^{D}\sum_{t=1}^{T}\sum_{i=1}^{N} \frac{dE_q[\log \beta_{k,v}]}{d\tilde{\beta}_{k,v}} I[x_{d,t,i} = v]\tilde{z}^k_{d,t,n} + \\ &(\omega_v - 1)\frac{dE_q[\log \beta_{k,v}]}{d\tilde{\beta}_{k,v}} - (\tilde{\beta}_{k,v} - 1)\frac{dE_q[\log \beta_{k,v}]}{d\tilde{\beta}_{k,v}} = 0 \end{aligned} \tag{11}$$

where the last equation uses the fact that:

$$\begin{aligned} \frac{d \log \Gamma(\tilde{\beta}_{k,v})}{d\tilde{\beta}_{k,v}} &= \Psi(\tilde{\beta}_{k,v}) \\ E_q[\log \beta_{k,v}] &= \Psi(\tilde{\beta}_{k,v}) - \Psi(\sum_{i=1}^{V} \tilde{\beta}_{k,i}) \end{aligned} \tag{12}$$

Hence, the update of the global parameters $\boldsymbol{\tilde{\beta}_k}$ while holding the other parameters fixed can be done as follow:

$$\boldsymbol{\tilde{\beta}_k} = \boldsymbol{\omega} + \sum_{d=1}^{D}\sum_{t=1}^{T}\sum_{i=1}^{N} \tilde{z}^k_{d,t,i}\boldsymbol{x_{d,t,i}} \quad \forall k \in \{1, ...K\} \tag{13}$$

where the notation $\boldsymbol{x_{d,t,i}}$ is overloaded to represent a vector whose $x_{d,t,i}$ element is equal to one and all the rest are zeros. Note that this update is analogous to that of LDA. As a function of the dynamic-related variational parameters $\boldsymbol{\tilde{A}}$

and $\tilde{\boldsymbol{A}}_{\mathbf{0}}$, we can write the objective as:

$$\mathcal{L}(\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{A}}_{\mathbf{0}}) = E_q[\log p(\boldsymbol{A}, \boldsymbol{A}_{\mathbf{0}}|\boldsymbol{y}, \boldsymbol{\alpha})] - E_q[\log q(\boldsymbol{A}, \boldsymbol{A}_{\mathbf{0}}|\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{A}}_{\mathbf{0}})] + const \tag{14}$$

$$E_q[\log p(\boldsymbol{A}, \boldsymbol{A}_{\mathbf{0}}|\boldsymbol{y}, \boldsymbol{\alpha})] = E_q[\log p(\boldsymbol{y}|\boldsymbol{A}, \boldsymbol{A}_{\mathbf{0}})] + E_q[\log p(\boldsymbol{A}, \boldsymbol{A}_{\mathbf{0}}|\boldsymbol{\alpha})] + const \tag{15}$$

$$\begin{aligned} E_q[\log p(\boldsymbol{y}|\boldsymbol{A}, \boldsymbol{A}_{\mathbf{0}})] = \sum_{d=1}^{D} \Bigg(& \sum_{l=1}^{P} E_q[\log A_{0,l}] q(y_{d,1} = l|\boldsymbol{a}_{\boldsymbol{d},\mathbf{0}}, \boldsymbol{l}_{\boldsymbol{d}}) \\ & + \sum_{t=1}^{T-1} \sum_{i=1}^{P} \sum_{j=1}^{P} E_q[\log A_{i,j}] q(y_{d,t} = i, y_{d,t+1} = j|\boldsymbol{a}_{\boldsymbol{d}}, \boldsymbol{l}_{\boldsymbol{d}}) \Bigg) \end{aligned} \tag{16}$$

$$E_q[\log p(\boldsymbol{A}, \boldsymbol{A}_{\mathbf{0}}|\boldsymbol{\alpha})] = \sum_{i=1}^{P} \Bigg(\sum_{j=1}^{P} (\alpha_j - 1) E_q[\log A_{i,j}] + (\alpha_{0i} - 1) E_q[\log A_{0i}] \Bigg) \tag{17}$$

$$\begin{aligned} E_q[\log q(\boldsymbol{A}, \boldsymbol{A}_{\mathbf{0}}|\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{A}}_{\mathbf{0}})] = \sum_{p=1}^{P} \Bigg(& \log \Gamma(\sum_{i=1}^{P} \tilde{A}_{p,i}) - \sum_{i=1}^{P} \log \\ \Gamma(\tilde{A}_{p,i}) + \sum_{i=1}^{P} (\tilde{A}_{p,i} - 1) E_q[\log A_{p,i}] \Bigg) & + \log \Gamma(\sum_{i=1}^{P} \tilde{A}_{0i}) \\ - \sum_{i=1}^{P} \log \Gamma(\tilde{A}_{0i}) & + \sum_{i=1}^{P} (\tilde{A}_{0i} - 1) E_q[\log A_{0i}] \end{aligned} \tag{18}$$

By taking the derivative of the objective function with respect to $\tilde{A}_{i,j}$ and set it to zero, we obtain the following:

$$\begin{aligned} \frac{d\mathcal{L}(\tilde{\boldsymbol{A}})}{d\tilde{A}_{i,j}} = 0 \implies & \sum_{d=1}^{D} \sum_{t=1}^{T-1} \frac{dE_q[\log A_{i,j}]}{d\tilde{A}_{i,j}} q(y_{d,t} = i, y_{d,t+1} = j|\boldsymbol{a}_{\boldsymbol{d}}, \boldsymbol{l}_{\boldsymbol{d}}) \\ + (\alpha_j - 1) \frac{dE_q[\log A_{i,j}]}{d\tilde{A}_{i,j}} & - (\tilde{A}_{i,j} - 1) \frac{dE_q[\log A_{i,j}]}{d\tilde{A}_{i,j}} = 0 \end{aligned} \tag{19}$$

Hence, the update of the global parameters $\tilde{A}_{i,j}$ while holding the other parameters fixed can be done as follow:

$$\tilde{A}_{i,j} = \alpha_j + \sum_{d=1}^{D} \sum_{t=1}^{T-1} q(y_{d,t} = i, y_{d,t+1} = j | \boldsymbol{a_d}, \boldsymbol{l_d}) \tag{20}$$

It can be noticed that this update is analogous to that of HMM [42]. Using HMM message passing recursions known as Forward-backward algorithm, $q(y_{d,t} = i, y_{d,t+1} = j | \boldsymbol{a_d}, \boldsymbol{l_d})$ can be computed in $O(TP^2)$ time as follows:

$$q(y_{d,t} = i, y_{d,t+1} = j | \boldsymbol{a_d}, \boldsymbol{l_d}) = f_{d,t,i} a_{d,i,j} \tilde{l}_{d,t+1,j} b_{d,t+1,j} / Z \tag{21}$$

where $f_{d,t,i}$ is the forward message at time t and $b_{d,t,j}$ is the backward one:

$$\begin{aligned} f_{d,t,i} &= \sum_{j=1}^{P} f_{d,t-1,j} a_{d,j,i} \tilde{l}_{d,t,i} \qquad f_{d,1,i} = a_{d,0,i} \\ b_{d,t,i} &= \sum_{j=1}^{P} a_{d,i,j} \tilde{l}_{d,t+1,j} b_{d,t+1,j} \qquad b_{d,T,i} = 1 \end{aligned} \tag{22}$$

Z is the normalisation constant. Note that performing inference for long time series (high T) is computationally intractable given the high-sampling rate of our data. Hence, the significance of combining the temporal ordering power of dynamic Bayesian networks with the automatic clustering power of hierarchical Bayesian models. By taking the derivative of $\mathcal{L}(\tilde{\boldsymbol{A}})$ with respect to $\tilde{A}_{0,j}$ and following the same pattern as Eq. (19), we obtain the following:

$$\begin{aligned} \tilde{A}_{0,j} &= \alpha_{0j} + \sum_{d=1}^{D} q(y_{d,1} = j | \boldsymbol{a_{d,0}}, \boldsymbol{l_d}) \\ &= \alpha_{0j} + \sum_{d=1}^{D} a_{d,0,j} b_{d,1,j} / Z \end{aligned} \tag{23}$$

As a function of the pattern-related variational parameters $\tilde{\boldsymbol{L}}$, we can write the objective as:

$$\mathcal{L}(\tilde{\boldsymbol{L}}) = E_q[\log p(\boldsymbol{L}|\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{\sigma})] - E_q[\log q(\boldsymbol{L}|\tilde{\boldsymbol{L}})] + const \tag{24}$$

$$E_q[\log p(\boldsymbol{L}|\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{\sigma})] = E_q[\log p(\boldsymbol{z}|\boldsymbol{L}, \boldsymbol{y})] + E_q[\log p(\boldsymbol{L}|\boldsymbol{\sigma})] \tag{25}$$

$$\begin{aligned} E_q[\log p(\boldsymbol{z}|\boldsymbol{L}, \boldsymbol{y})] = \sum_{d=1}^{D}\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{P}\sum_{k=1}^{K} E_q[\log L_{j,k}] \\ q(y_{d,t} = j|\boldsymbol{a_d}, \boldsymbol{a_{d,0}}, \boldsymbol{l_d})q(z_{d,t,i} = k|\tilde{\boldsymbol{z}}_{\boldsymbol{d,t,i}}) \end{aligned} \tag{26}$$

$$E_q[\log p(\boldsymbol{L}|\boldsymbol{\sigma})] = \sum_{j=1}^{P}\sum_{i=1}^{K}(\sigma_i - 1)E_q[\log L_{j,i}]$$

$$E_q[\log q(\boldsymbol{L}|\tilde{\boldsymbol{L}})] = \sum_{p=1}^{P}\left(\log\Gamma(\sum_{k=1}^{K}\tilde{L}_{p,k}) - \sum_{k=1}^{K}\log\Gamma(\tilde{L}_{p,k}) + \sum_{k=1}^{K}(\tilde{L}_{p,k} - 1)E_q[\log L_{p,k}]\right) \tag{27}$$

By taking the derivative with respect to $\tilde{L}_{p,k}$ and setting it to zeros, we obtain the following update:

$$\tilde{L}_{p,k} = \sigma_k + \sum_{d=1}^{D}\sum_{t=1}^{T}\sum_{i=1}^{N} q(y_{d,t} = p|\boldsymbol{a_d}, \boldsymbol{a_{d,0}}, \boldsymbol{l_d})q(z_{d,t,i} = k|\tilde{\boldsymbol{z}}_{\boldsymbol{d,t,i}}) \tag{28}$$

$$q(z_{d,t,i} = k|\tilde{\boldsymbol{z}}_{\boldsymbol{d,t,i}}) = \tilde{z}^{k}_{d,t,i} \tag{29}$$

$$q(y_{d,t} = p|\boldsymbol{a_d}, \boldsymbol{a_{d,0}}, \boldsymbol{l_d}) = f_{d,t,p}b_{d,t,p}/Z \tag{30}$$

We can see that the update here involves additional term (Eq. (29)) to those of the HMM update that reflect the influence of the LDA-part on the HMM-part.

3.2. Local Parameters

In this section, we derive the coordinate update for the local parameters $\{\tilde{\boldsymbol{z}}, \boldsymbol{a_0}, \boldsymbol{a}, \tilde{\boldsymbol{l}}\}$ of the variational distribution. As a function of the variational parameters $\tilde{\boldsymbol{z}}$, we can write the objective as:

$$\mathcal{L}(\tilde{\boldsymbol{z}}) = E_q[\log p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\beta}, \boldsymbol{y}, \boldsymbol{L})] - E_q[\log q(\boldsymbol{z}|\tilde{\boldsymbol{z}})] + const \tag{31}$$

$$E_q[\log p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\beta}, \boldsymbol{y}, \boldsymbol{L})] = E_q[\log p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\beta})] + E_q[\log p(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{L})] \tag{32}$$

$$E_q[\log p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\beta})] = \sum_{d=1}^{D}\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{k=1}^{K} \tilde{z}^k_{d,t,i} E_q[\log \beta_{k,x_{d,t,i}}] \tag{33}$$

$$E_q[\log p(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{L})] = \sum_{d=1}^{D}\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{p=1}^{P}\sum_{k=1}^{K} \tilde{z}^k_{d,t,i} f_{d,t,p} b_{d,t,p} E_q[\log L_{p,k}] \tag{34}$$

$$E_q[\log q(\boldsymbol{z}|\tilde{\boldsymbol{z}})] = \sum_{d=1}^{D}\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{p=1}^{P}\sum_{k=1}^{K} \tilde{z}^k_{d,t,i} \log \tilde{z}^k_{d,t,i} \tag{35}$$

By taking the derivative with respect to $\tilde{z}^k_{d,t,i}$ and setting it to zeros, we obtain the following update:

$$\tilde{z}^k_{d,t,i} \propto \exp\left(\Psi(\tilde{\boldsymbol{\beta}}_{\boldsymbol{k}})^T \boldsymbol{x_{d,t,i}} - \Psi(\sum_{i=1}^{V} \tilde{\beta}_{k,i}) + \sum_{p=1}^{P} f_{d,t,p} b_{d,t,p} \Psi(\tilde{L}_{p,k})\right) \tag{36}$$

The first two terms of Eq. (36) show the LDA contribution to the updates while the last term is the HMM's. ELBO can be written as a function of the variational parameters $\{\boldsymbol{a_0}, \boldsymbol{a}, \tilde{\boldsymbol{l}}\}$ as follows:

$$\mathcal{L}(\boldsymbol{a_0}, \boldsymbol{a}, \tilde{\boldsymbol{l}}) = E_q[\log p(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{L}, \boldsymbol{A_0}, \boldsymbol{A})] - E_q[\log q(\boldsymbol{y}|\tilde{\boldsymbol{l}}, \boldsymbol{a}, \boldsymbol{a_0})] + const \tag{37}$$

$$\mathcal{L}(\boldsymbol{a_0}, \boldsymbol{a}, \tilde{\boldsymbol{l}}) = \sum_{d=1}^{D} \Bigg(E_q \Bigg[\log A_{0y_{d,1}} + \sum_{t=1}^{T-1} \log A_{y_{d,t},y_{d,t+1}} + \sum_{t=1}^{T} \sum_{i=1}^{N} \log L_{y_{d,t},z_{d,t,i}} \Bigg] - E_q \Bigg[\log a_{0d,y_{d,1}} + \sum_{t=1}^{T-1} \log a_{d,y_{d,t},y_{d,t+1}} + \sum_{t=1}^{T} \log \tilde{l}_{d,t,y_{d,t}} \Bigg] \Bigg) \tag{38}$$

By taking the derivative with respect to $\{a_{0d,p}, a_{d,i,j}, \tilde{l}_{d,t,p}\}$, we obtain the following updates:

$$\begin{aligned} a_{0d,p} &\propto \exp E_q[\log A_{0p}] \\ &= \exp \Bigg(\Psi(\tilde{A}_{0p}) - \Psi(\sum_{i=1}^{P} \tilde{A}_{0i}) \Bigg) \end{aligned} \tag{39}$$

$$\begin{aligned} a_{d,i,j} &\propto \exp E_q[\log A_{i,j}] \\ &= \exp \Bigg(\Psi(\tilde{A}_{i,j}) - \Psi(\sum_{j=1}^{P} \tilde{A}_{i,j}) \Bigg) \end{aligned} \tag{40}$$

$$\begin{aligned} \tilde{l}_{d,t,p} &\propto \exp \sum_{i=1}^{N} \sum_{k=1}^{K} \tilde{z}^{k}_{d,t,i} E_q[\log L_{p,k}] \\ &= \exp \sum_{i=1}^{N} \sum_{k=1}^{K} \tilde{z}^{k}_{d,t,i} \Bigg(\Psi(\tilde{L}_{p,k}) - \Psi(\sum_{j=1}^{K} \tilde{L}_{p,j}) \Bigg) \end{aligned} \tag{41}$$

We can clearly see the resemblance of these updates to those of HMM SVI proposed in [42]. The slight divergence is caused by the fact that here the observations are replaced by many latent variables $z_{d,t,n}$.

3.3. Algorithm

Because LDA-HMM is member of the family of models presented in [18], its SVI can be derived from VI following the same steps. Only the updates of the global variational parameters are modified so that no need to iterate D times over the observations or local parameters before updating the global ones.

$$\tilde{\boldsymbol{\beta}}_{\boldsymbol{k}} = \boldsymbol{\omega} + D \sum_{t=1}^{T} \sum_{i=1}^{N} \tilde{z}^{k}_{d,t,i} \boldsymbol{x_{d,t,i}} \quad \forall k \in \{1, ...K\} \tag{42}$$

$$\tilde{A}_{i,j} = \alpha_j + D \sum_{t=1}^{T-1} f_{d,t,i} a_{d,i,j} \tilde{l}_{d,t+1,j} b_{d,t+1,j}/Z \tag{43}$$

$$\tilde{A}_{0,j} = \alpha_{0j} + D a_{d,0,j} b_{d,1,j}/Z \tag{44}$$

$$\tilde{L}_{p,k} = \sigma_k + D \sum_{t=1}^{T} \sum_{i=1}^{N} \tilde{z}^{k}_{d,t,i} f_{d,t,p} b_{d,t,p}/Z \tag{45}$$

where d is sampled uniformly from $\{1, ...D\}$. In the following, we present SVI algorithm of LDA-HMM which includes updates equivalent to following noisy estimates of the natural gradient of the ELBO.

4. Experiments

In this section, we will first introduce the experimental data DBN-LDA-HMM will be tested on along with details about the data pre-processing stages. Next, we define the experimental settings, introduce the evaluation strategy, present and discuss the results. In order to provide some comparisons, we derive two new simpler unsupervised pattern mining methods. For the first one, the DBN-like and HMM-like parts are omitted resulting in LDA with continuous observation. In order to process continuous observations, the dirichlet distribution of LDA is replaced by Gaussian one resulting in Gaussian Latent Dirichlet Allocation (GLDA) [38]. We also derive DBN-LDA model where the HMM-part is omitted [39]

Algorithm 1 Deep Online Hierarchical Dynamic Unsupervised Pattern mining for energy consumption behaviour

```
Input: raw-data window length, $R$; preprocessed-data window length, $N$; length of time series, $T$; number of components, $K$; number of patterns, $P$: total number of iterations, $C$; learning rate parameters, $\kappa$ and $\tau_0$; hyper-parameters, $\boldsymbol{\alpha}$, $\boldsymbol{\alpha_0}$, $\boldsymbol{\omega}$ and $\boldsymbol{\sigma}$.
Initialisation: variational parameters: $\{\tilde{\boldsymbol{\beta}}_{\boldsymbol{k}}\}_{k=1}^{K}$, $\{\tilde{\boldsymbol{A}}_{\boldsymbol{p}}, \tilde{\boldsymbol{L}}_{\boldsymbol{p}}\}_{p=1}^{P}$ and $\tilde{\boldsymbol{A}}_{\boldsymbol{0}}$, learning rates $\{\rho_c = f(c, \tau_0, \kappa)\}_{c=0}^{C}$ (see [43]).
for $c = 0, 1, 2, ...C - 1$ do
    Read sequentially $NT$ raw data windows of length $R$.
    Extract features using the pre-trained DBN (see Sec. 4) for each window
    Form windows of time series data points (actual input) of length $NT$ in the new feature space $(\{\{\boldsymbol{x}_{d,t,n}\}_{n=1}^{N}\}_{t=1}^{T})$
    Initialise $\{\boldsymbol{a_{0d}}, \{\boldsymbol{a_{d,p}}, \tilde{\boldsymbol{l}}_{\boldsymbol{d,p}}\}_{p=1}^{P}\}$
    repeat
        Compute local variational parameters $\{\{\{\tilde{\boldsymbol{z}}_{\boldsymbol{d,t,i}}\}_{t=1}^{T}\}_{i=1}^{N}\}_{k=1}^{K}$ (see Eq. (36))
        Update local variational parameters $\{\boldsymbol{a_{0d}}, \{\boldsymbol{a_{d,p}}, \tilde{\boldsymbol{l}}_{\boldsymbol{d,p}}\}_{p=1}^{P}\}$ (see Eq. (39), Eq. (40) and Eq. (41))
    until local parameters converge
    Compute intermediate global variational parameters $\{\tilde{\boldsymbol{\beta}}'_{\boldsymbol{k}}\}_{k=1}^{K}$, $\{\tilde{\boldsymbol{A}}'_{\boldsymbol{p}}, \tilde{\boldsymbol{L}}'_{\boldsymbol{p}}\}_{p=1}^{P}$ and $\tilde{\boldsymbol{A}}'_{\boldsymbol{0}}$ (see Eq. (42), Eq. (43), Eq. (44) and Eq. (45))
    Update the current estimate of the global variational parameters: $\{\tilde{\boldsymbol{\beta}}_{\boldsymbol{k}} = (1 - \rho_c)\tilde{\boldsymbol{\beta}}_{\boldsymbol{k}} + \rho_c \tilde{\boldsymbol{\beta}}'_{\boldsymbol{k}}\}_{k=1}^{K}$, $\{\tilde{\boldsymbol{A}}_{\boldsymbol{p}} = (1 - \rho_c)\tilde{\boldsymbol{A}}'_{\boldsymbol{p}} + \rho_c \tilde{\boldsymbol{A}}_{\boldsymbol{p}}, \tilde{\boldsymbol{L}}_{\boldsymbol{p}} = (1 - \rho_c)\tilde{\boldsymbol{L}}'_{\boldsymbol{p}} + \rho_c \tilde{\boldsymbol{L}}_{\boldsymbol{p}}\}_{p=1}^{P}$ and $\tilde{\boldsymbol{A}}_{\boldsymbol{0}} = (1 - \rho_c)\tilde{\boldsymbol{A}}'_{\boldsymbol{0}} + \rho_c \tilde{\boldsymbol{A}}_{\boldsymbol{0}}$
end for
```

4.1. Datasets

The real-world multi-source utility usage data used here is provided by ETI[1]. The data includes electricity signals (voltage and current signals) sampled at high sampling rate around 205 kHz, water and gas consumption sampled at low sampling rate. The data also contains other sensors measurements collected from the Home Energy Monitoring System (HEMS). In this study we will use 4Tb of utility usage data collected from one house over one month. This data has been recorded into three different formats. Water data is stored in text files with sampling rate of 10 seconds and is synchronised to Network Time

[1] Energy Technologies Institute: http://www.eti.co.uk/

Table. 2 Characteristics of the data

Data	Range	Resolution	Measurement frequency	Total duration
Mains Voltage	-500V to +500V	62mV	4.88s	1 months
Mains Current	-10A to +10A	1.2mA	4.88s	1 months
Water Flow Volume	0 to 100L per min	52.4 pulses per litre	10s	1 months
Room Air Temperature	0 to 40 DegC	0.1 DegC	Once every minute	1 months
Room Relative Humidity	0 to 95%	0.1 %	Once every 5 minutes	1 months
Hot Water Feed Temperature	DegC	0.1 DegC	Once every 5 minutes	1 months
Boiler: Water Temperature (Input)	0 to 85 DegC	0.1 DegC	Once every 5 minutes	1 months
Boiler: Water Temperature (Output)	DegC	0.1 DegC	Once every 5 minutes	1 months
Household: Mains Cold Water Inlet Temperature	DegC	0.1 DegC	Once every 5 minutes	1 months
Gas Meter Reading	Metric Meter	0.01m3	Once every 15 minutes	1 months
Radiator Temperature	DegC	0.1 DegC	Once every 5 minutes	1 months
Radiator Valve	0 to 100%	50%	Once every 5 minutes	1 months
Boiler Firing Switch	Boolean	None	Once every 5 minutes	1 months

Protocol (NTP) approximately once per month. Electricity data is stored in wave files with sampling rate of 4.88 s and is synchronised to NTP every 28min 28sec. HEMS data is stored in a Mongo database with sampling rates differing according to the type of the data and sensors generating it (see Tab. 2).

4.1.1. Data Pre-processing

In order to use raw utility data, a number of pre-processing steps are required. We implemented a Python code that reads the data from these different sources, synchronises its time-stamps to NTP time-stamps, extracts features and aligns the data samples to one time-stamp by measurement. For water data, the PC clock time-stamps of samples within each month are synchronised

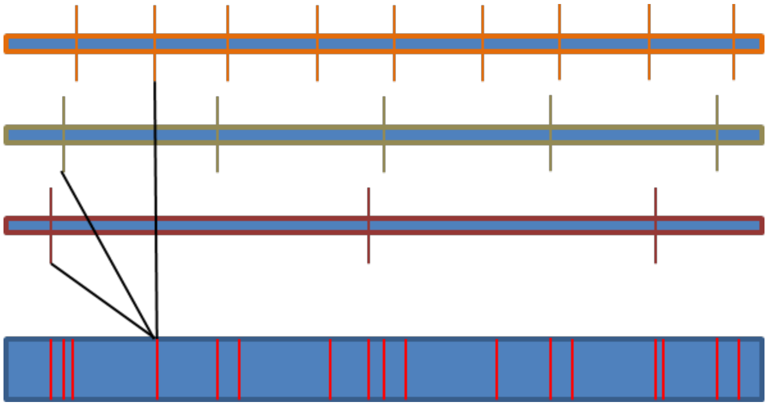


Figure. 4 Alignment of the data

to NTP time-stamp. The synchronisation is done as follows:

$$timestampNTP(i) = timestampsclock(i) + i\frac{Total\_Time\_Shift}{Number\_of\_Samples} \quad (46)$$

In this equation, we assume that the total shift (between NTP and PC clock) can be distributed over the samples in one month. Similarly, Electricity data samples' time-stamps are synchronised to NTP time-stamps. The shift is distributed over 28 minutes and 28 seconds.

$$timestampNTP(i) = timestampsclock(i) + i\frac{Total\_Time\_Shift}{Number\_of\_Samples} \quad (47)$$

The time-stamps of HEMS data were collected using NTP and so no synchronisation is required. Having all data samples synchronised to the same reference (NTP), we align the samples to the same time-stamps. The alignment strategy is shown in Fig. 4 where the union of all aligned data samples is stored in one matrix. Each row of this matrix includes a time-stamp and the corresponding values of the sensors. If for some sensors, there are no measurements taken at the time-stamp, the values measured at the previous time stamp are taken. The aligned data samples are the input of the feature extraction model. Pushed by the complexity of the mining task and motivated by the informativeness and simplicity of the water and sensors data, at this stage, we only extract features from the electricity data over time windows of 1 second. These features are then aligned following the same process described earlier. We propose two different features extraction mechanisms. For GLDA, we use NILM features known to work well with NILM supervised learning. For DBN-LDA and DBN-LDA-HMM, we use deep learning to learn the features. Details about the features extraction mechanism and DBN are provided in App. 6

4.2. Experimental Settings

In this section, we focus on the experiments performed on the pre-processed data, where the online LDA-HMM is applied on the features extracted by DBN. We also provide some comparisons with GLDA and DBN-LDA.

In all experiments, the number of global components (i.e., patterns and appliances related components) K and P are fixed to 30 and 20. The DBN's input granularity is set to 1 second (See App. 6) for both DBN-LDA and DBN-LDA-HMM. Given the data granularity, this leads to raw-data windows length $R = 204911$. The pre-processed-data windows N is set so that the granularity of the patterns is 30 minutes, hence N is equal to $30 * 60$. The length of time series T (DBN-LDA-HMM exclusive parameter) is fixed to span a whole day (i.e., 24 hours), hence T is equal to 48.

we use the perplexity to measure the model fitness to the data. It is defined as the reciprocal geometric mean of the inverse marginal probability of the input in the held-out test set. Since perplexity cannot be computed directly, a lower bound on it is derived in a similar way to the one in [22]. This bound is used as a proxy for the perplexity. To set the Dirichlet distribution hyper-parameters, we ran experiments with $\alpha_0 = \{0.001, 0.01, 0.1, 1\}$, $\alpha = \{0.001, 0.01, 0.1, 1\}$, $\sigma = \{0.001, 0.01, 0.1, 1\}$, $\omega = \{0.001, 0.01, 0.1, 1\}$ on held-out data that is not part of the one month one used for training and testing (see Sec. 4.1). We then chose the parameters that provided the highest preplexity $\{\alpha_0 = 0.1, \alpha = 0.1, \sigma = 0.1, \omega = 0.1\}$. We also evaluated a range of settings of the learning parameters on DBN-LDA: $\kappa \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$ (learning factor), $\tau_0 \in \{1, 64, 256, 1024\}$ (learning delay) and batch size $BS \in \{1, 4, 8\}$ on the held-out data, where the

Table. 3 Parameter settings

Batch size: BS	1	4	8
Learning factor: κ	0.7	0.5	0.5
Learning delay: τ_0	256	64	64
Perplexity	334	333	350

parameters κ and τ_0, defined in [43], control the learning step-size ρ_j. In these experiments, we omitted HMM as we are interested in LDA parameters and to speed up the experiments. Table 3 summarises the best settings of each batch size along with the perplexity obtained on the held-out data.

The obtained results show that the perplexity for different parameters settings are similar. However, the computation complexity increases with the size of the batch. Hence, we set the batch size to 1, where the best learning parameters are $\kappa = 0.7$ and $\tau = 256$. In the following, we used the data collected during the first two weeks for testing and that of the last two weeks for training. All experiments are run 30 times.

4.3. Global Components

The learned components by DBN-LDA-HMM correspond to clusters in DBN's output space and patterns of energy consumption activities. The clusters in DBN's output space are represented by multinomial distribution over the discrete DBN's outputs. Different mixing proportions of these clusters represent the patterns components. Hence, to visualise these components, we plot gray-scale images of these components where black colour indicates zeros-probability and white colour indicates one-probability. Figure 5 shows the aforementioned clusters where x-axis corresponds to different clusters and y-axis represents the discrete DBN's outputs. Figure 6 shows the patterns (x-axis) represented by different cluster's proportions (y-axis), where each square indicates the probability of a cluster with black colour indicates zeros-probability and white colour indicates one-probability.

Figure 5 shows that DBNs outputs at most of the dimensions are zeros (covered in black). Hence, we conclude that the deep learning successfully extract discriminative features with much lower dimensions than the input data. Par-

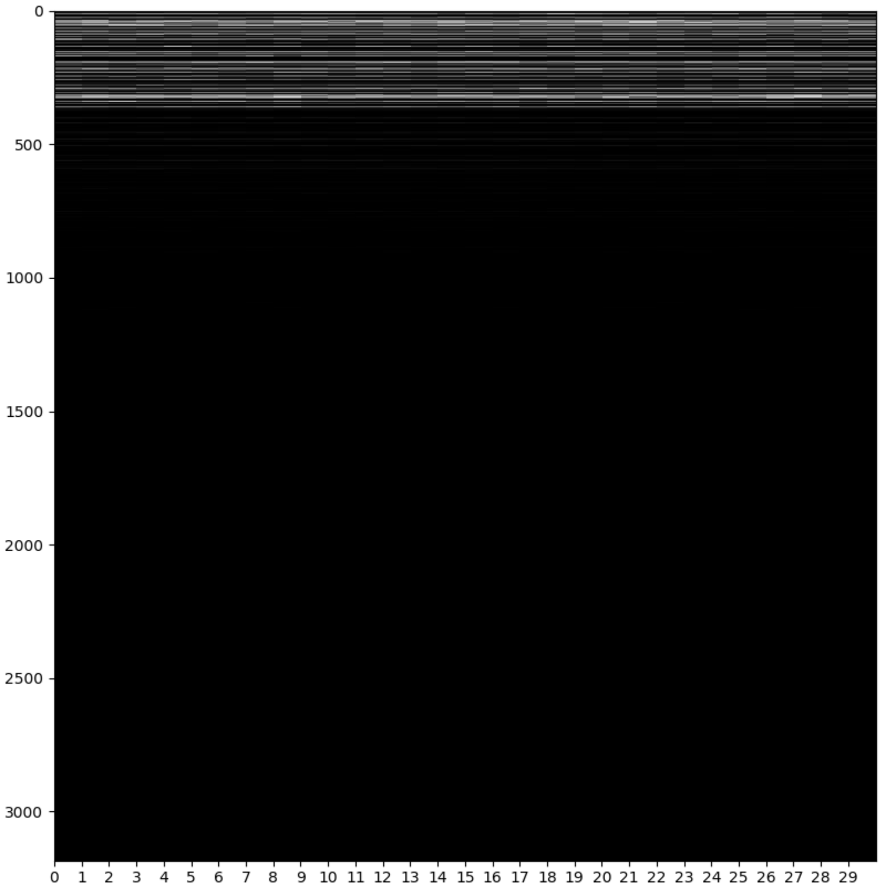


Figure. 5 Clusters in DBN's output space

ticularly, it can be seen from Fig. 5 that only around 350 dimensions of DBN's output represent the appliance related components that summarise the data. Hence, different combination in this 350 dimensions form the clusters whose mixtures represents patterns. In analogy to topics model, these are the words composing the topics forming the documents. This observation shows that DBN has managed to reduce the real high dimensional input space to discrete lower dimensional output space where countably small number of points represents most of the input signal over 1 second granularity. Hence, we expect these points to have strong relation with appliances usage. This is supported by the

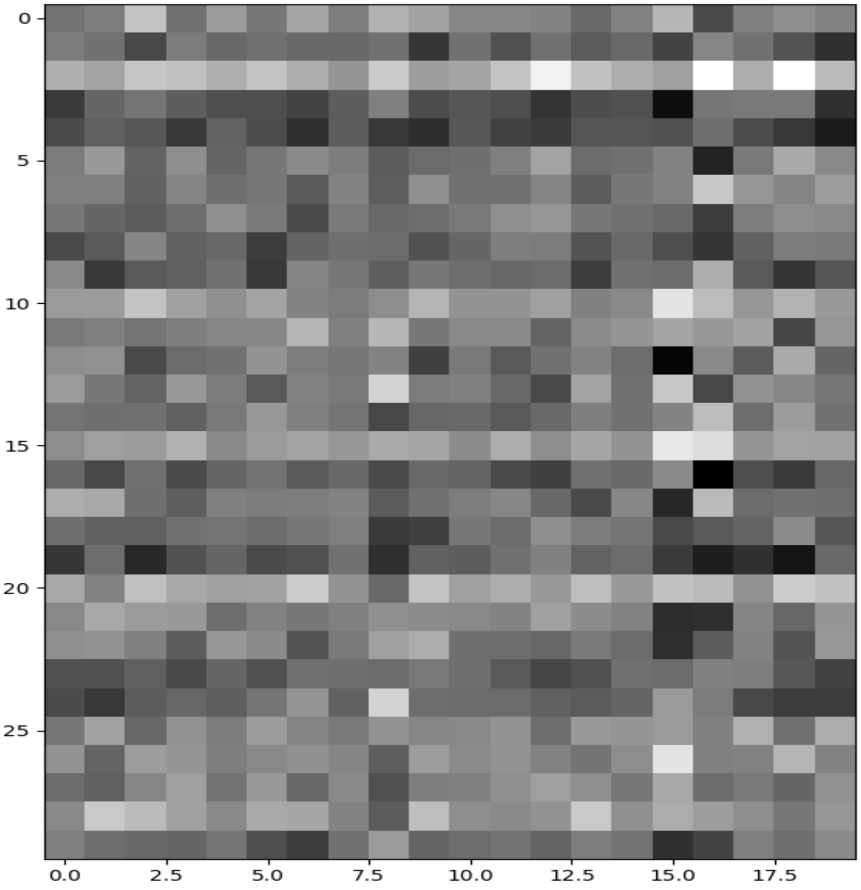


Figure. 6 Patterns of energy consumption activities

few light coloured points appearing with each component meaning that different clusters (appliance-related components) are mainly composed of these points.

The sparse light coloured rectangle in Fig. 6 implies that the majority of patterns consists mainly of few appliance-related components. For example, the main components for breakfast patterns will relate to cooking and heating appliances such as the hob and the oven. Figure 6 also shows few horizontal strips of light colour, for example at components 2, 10,15 and 20. These components therefore appears in most patterns meaning that they may belong to appliances like lambs which are used in different activities. In the following, we propose two evaluation methods to support our claims about the relation between clusters and appliance, patterns and activities.

4.4. Evaluation and Analysis

In order to investigate the quality of the results, we study the regularity of the mined patterns by matching them across similar periods of time. For instance, it is expected that similar patterns will emerge in specific times like breakfast in every morning, watching TV in the evening, etc. This regularity can also be seen across days, for instance, consumption behaviour during working days is different from that during the weekend. Hence, it is interesting to understand how such patterns occur as regular events.

We also provide a quantitative evaluation of the algorithm by proposing a mapping method that reveals the specific energy consumed from the inferred patterns within the patterns' granularity (fixed to 30 minutes). By doing so, we can evaluate numerically the consistency between energy consumption and the extracted patterns. This is achieved by fitting a regression model to the energy consumption over the K components (clusters in DBN's output space):

$$A\boldsymbol{w} = \boldsymbol{b} \tag{48}$$

where $\boldsymbol{w}$ is a vector expressing energy consumption associated with components, $\boldsymbol{b}$ is a vector representing consumption within the patterns' granularity and A is the matrix of the components proportions obtained by DBN-LDA-HMM. The matrix A is the result of $A1A2$, where $A1$ represents the patterns proportions within the patterns' granularity and $A2$ is the clusters proportions within the patterns. This technique will also allow numerically checking the predicted consumption against the real consumption.

4.4.1. Pattern Regularity

Using the optimal parameters' setting, we examine in the following the regularity of the mined patterns. To do that, we use the first two weeks of the data for testing. To study the regularity of the energy consumption behaviour of the

Table. 4 Patterns dissimilarity matrix for DBN-LDA-HMM

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Mon	0.0052	0.0048	0.0064	0.0069	0.007	0.0089	0.0091
Tue	0.0048	0.0049	0.0055	0.006	0.0064	0.0086	0.0095
Wed	0.0064	0.0055	0.0049	0.007	0.0071	0.0081	0.0091
Thu	0.0069	0.006	0.007	0.0058	0.0074	0.0085	0.009
Fri	0.007	0.0064	0.0071	0.0073	0.0068	0.008	0.0096
Sat	0.0089	0.0086	0.0081	0.0086	0.008	0.0082	0.0080
Sun	0.0091	0.0095	0.0091	0.0093	0.0096	0.0080	0.0088

Table. 5 Patterns dissimilarity matrix for DBN-HMM

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Mon	0.0078	0.0071	0.0096	0.0104	0.0105	0.0133	0.0136
Tue	0.0071	0.0072	0.0082	0.009	0.0096	0.0129	0.0142
Wed	0.0096	0.0082	0.0073	0.0105	0.0107	0.0121	0.0137
Thu	0.0104	0.009	0.0105	0.0087	0.0111	0.0127	0.0136
Fri	0.0105	0.0096	0.0107	0.0111	0.0102	0.012	0.0144
Sat	0.0133	0.0129	0.0121	0.0127	0.012	0.0124	0.0121
Sun	0.0136	0.0142	0.0137	0.0136	0.0144	0.0121	0.012

residents, we compare the mined patterns across different days of the testing period. The patterns of day d is computed as follow:

$$p(y_{d,t}|\{\{\boldsymbol{x_{d,t}}\}_{t=1}^{T}\}_{d=1}^{D}, \boldsymbol{\alpha}, \boldsymbol{\alpha_0}, \boldsymbol{\sigma}, \boldsymbol{\omega}) \tag{49}$$

This distribution is intractable and the proxy approximation shown in Eq. 5 is used instead. The similarity of the patterns across the two weeks are computed as follows:

$$\begin{aligned} similarity(day_a, day_b) = \frac{1}{P*T}\sum_{t=1}^{T}\sum_{p=1}^{P}\Bigg| p(y_{a,t} = p|\{\{\boldsymbol{x_{d,t}}\}_{t=1}^{T} \\ \}_{d=1}^{D}, \boldsymbol{\alpha}, \boldsymbol{\alpha_0}, \boldsymbol{\sigma}, \boldsymbol{\omega}) - p(y_{b,t} = p|\{\{\boldsymbol{x_{d,t}}\}_{t=1}^{T}\}_{d=1}^{D}, \boldsymbol{\alpha}, \boldsymbol{\alpha_0}, \boldsymbol{\sigma}, \boldsymbol{\omega})\Bigg| \end{aligned} \tag{50}$$

Table 4 shows the per-day similarity. It can be clearly seen from the table that there is regular patterns across the same days from two different weeks. That is, similar energy consumption patterns appear across these days. This similarity

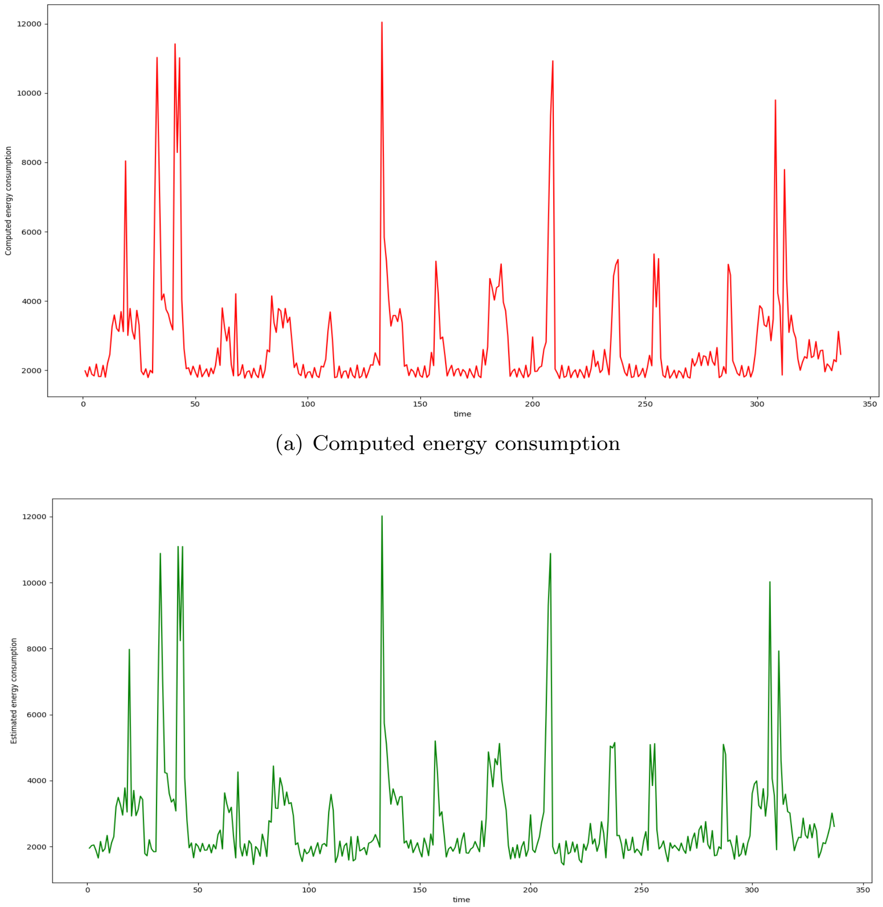
Table. 6 Patterns dissimilarity matrix for GLDA

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Mon	0.0093	0.0085	0.0115	0.0125	0.0126	0.0159	0.0163
Tue	0.0085	0.00864	0.00984	0.0108	0.0115	0.0155	0.017
Wed	0.0115	0.00984	0.0084	0.0125	0.0127	0.0145	0.0164
Thu	0.0125	0.0108	00.0125	0.0104	0.0133	0.0152	0.0163
Fri	0.0126	0.0115	0.0127	0.0133	0.0122	0.0144	0.0173
Sat	0.0159	0.0155	0.0145	0.0152	0.0144	0.0149	0.0145
Sun	0.0163	0.017	0.0164	0.0163	0.0173	0.0145	0.0144

is a bit less for the weekend where more random activities could take place. Computing the similarity measure between week and weekend days confirms this observation. For instance, the similarity between first week's Monday and second week's Sunday is equal to 0.0093 which is much higher than that between the Mondays of the two weeks. In contrast, the similarity among working days are generally high. We perform similar regularity studies on GLDA and DBN-LDA to provide some comparisons. Table 5 and Tab. 6 show the per-day similarity of DBN-LDA and GLDA respectively. Comparing the results in Tab. 4 to that in Tab. 5 and Tab. 6, it is clearly noticeable that our method DBN-LDA-HMM was able to capture the patterns regularities better than both DBN-LDA and GLDA. We can also notice the improvement obtained by using deep learning (i.e., DBN) for features extraction compared to the engineering features approach taken by GLDA (see Tab. 5 and Tab. 6).

The captured regularity may be caused by regular user lifestyle leading to similar energy consumption behaviour within and across the weeks. Such regularity is violated in the weekend, as more random activities could take place. Having shown that there is some regularity in the mined patterns, it is more likely that specific energy consumption can be associated with each component. In the next section, we apply a regression method to map the patterns within the patterns' granularity (fixed to 30 minutes) to energy consumption. Thus, the parameters of interest are the energy consumption associated with the components. By attaching an energy consumption with each component, we can help validate the coherence of the extracted patterns and evaluate numerically the consistency between energy consumption and the extracted patterns which can be exploited to predict the load demand.

4.4.2. Energy Mapping



(a) Computed energy consumption

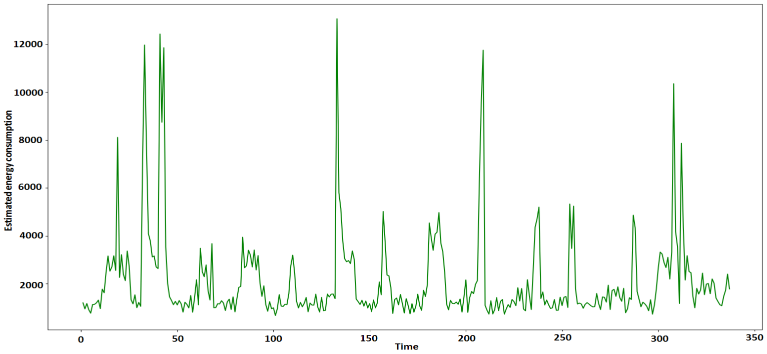
(b) Estimated energy consumption by DBN-LDA-HMM

Figure. 7 Evolution of the energy consumption over time

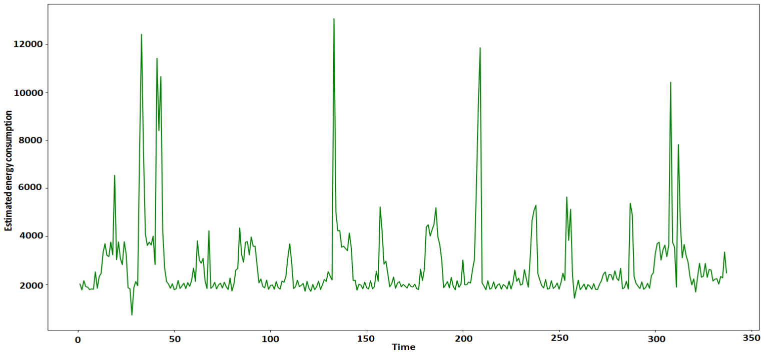
As shown in the previous section, DBN-LDA-HMM can express the energy consumption patterns by mixing multinomial distributions over mixture of components (clusters) summarising the data. Each component is a distribution over a high-dimensional feature space and understanding what it represents is not easy. Hence, we propose to associate consumption quantities to each component. Such association is motivated by the fact that an energy consumption pattern is normally governed by the usage of different appliances in the house. There should be a strong relation between components and appliances usage. Hence, a relation between components and energy consumption is plausible. Note that the best case scenario occurs if each component is associated with the usage of a specific appliance. Apart from the coherence study, associating energy consumption with each component can be used to predict the energy consumption demand. As explained in Sec. 4.4, we apply a simple least-square regression method to map the inferred patterns within the patterns' granularity (fixed to 30 minutes) to energy consumption. We train the regression model on one of the testing week and run the model on the other testing week. Figure 7 shows the energy consumption (in joules) along with the estimated consumption computed using the learned per-component consumption parameters.

The similarity between the estimated and computed energy consumption demonstrates that the DBN-LDA-HMM components express distinct usages of energy. Such distinction can be the result of the usage of different appliances likely having distinct energy consumption signatures. Thus, the proposed approach produces coherent and regular patterns that reflect the energy consumption behaviour and human activities. Note that it is possible that different patterns (or appliance usages) may have the same energy consumption and that might be one reason why both estimated and computed energy consumption in Fig. 7 are not fully the same.

Finally, we perform energy consumption estimation by GLDA and DBN-LDA. Figrue 8 shows that DBN-LDA provide better performance than GLDA which is inline with the pattern regularity study in Sec. 4.4.1. Although the estimated energy consumption by DBN-LDA and DBN-LDA-HMM seems close, DBN-LDA-HMM still shows better estimation. This supports the better performance of DBN-LDA-HMM shown with the pattern regularity study in Sec. 4.4.1.



(a) Estimated energy consumption by LDA-HMM



(b) Estimated energy consumption by GLDA

Figure. 8 Evolution of the energy estimation by DBN-LDA and GLDA over time

5. Conclusion

In this paper, we presented a novel approach to extract patterns of the users' consumption behaviour from data involving different utilities (e.g, electricity, water and gas) as well as some sensors measurements. DBN-LDA-HMM is fully unsupervised and the LDA-HMM component' training is done online which made it efficient for the fast learning of big data. To analyse the performance, we proposed a two-step evaluation that covers: patterns regularity and coherency. The experiments show that the proposed method is capable of extracting regular and coherent patterns that highlight energy consumption over time.

The main limitation of the proposed approach is the difficulty to provide standard quantitative empirical evaluation. In the future, we foresee four directions for research to improve the obtained results and provide more features: (i) involving some labelling in the data collection process and devise a new semi-supervised approach from DBN-LDA-HMM (ii) improving the scalability of the algorithm to learn from the whole data (whose size 80 terabytes) by applying asynchronous distributed inference which can be derived from [44], (iii) considering other houses by designing novel models that allow transfer learning and (iiii) involving active learning strategy with the new semi-supervised version to query users (residents) about their activities in order to guide the learning process when needed [45, 46].

Acknowledgment

This work was supported by the Energy Technology Institute (UK) as part of the project: *High Frequency Appliance Disaggregation Analysis (HFADA)*.

References

[1] A. Bulling, U. Blanke, B. Schiele, A tutorial on human activity recognition using body-worn inertial sensors, ACM Computing Surveys (CSUR) 46 (3) (2014) 33.

[2] R. Poppe, A survey on vision-based human action recognition, Image and vision computing 28 (6) (2010) 976–990.

[3] S. Wang, G. Zhou, A review on radio based activity recognition, Digital Communications and Networks 1 (1) (2015) 20–29.

[4] G. W. Hart, Nonintrusive appliance load monitoring, Proceedings of the IEEE 80 (12) (1992) 1870–1891.

[5] M. Zeifman, K. Roth, Nonintrusive appliance load monitoring: Review and outlook, IEEE Transactions on Consumer Electronics 1 (57) (2011) 76–84.

[6] A. Zoha, A. Gluhak, M. A. Imran, S. Rajasegarar, Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey, Sensors 12 (12) (2012) 16838–16866.

[7] J. Liang, S. K. Ng, G. Kendall, J. W. Cheng, Load signature studypart i: Basic concept, structure, and methodology, IEEE transactions on power Delivery 25 (2) (2010) 551–560.

[8] J. Z. Kolter, S. Batra, A. Y. Ng, Energy disaggregation via discriminative sparse coding, in: Advances in Neural Information Processing Systems, 2010, pp. 1153–1161.

[9] D. Srinivasan, W. Ng, A. Liew, Neural-network-based signature recognition for harmonic source identification, IEEE Transactions on Power Delivery 21 (1) (2006) 398–405.

[10] M. Berges, E. Goldman, H. S. Matthews, L. Soibelman, Learning systems for electric consumption of buildings, in: Computing in Civil Engineering (2009), 2009, pp. 1–10.

[11] A. G. Ruzzelli, C. Nicolas, A. Schoofs, G. M. O'Hare, Real-time recognition and profiling of appliances through a single electricity sensor, in: Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2010 7th Annual IEEE Communications Society Conference on, IEEE, 2010, pp. 1–9.

[12] J. Kelly, W. Knottenbelt, Neural nilm: Deep neural networks applied to energy disaggregation, in: Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments, ACM, 2015, pp. 55–64.

[13] Y.-X. Lai, C.-F. Lai, Y.-M. Huang, H.-C. Chao, Multi-appliance recognition system with hybrid svm/gmm classifier in ubiquitous smart home, Information Sciences 230 (2013) 39–55.

[14] R. Bonfigli, S. Squartini, M. Fagiani, F. Piazza, Unsupervised algorithms for non-intrusive load monitoring: An up-to-date overview, in: Environment and Electrical Engineering (EEEIC), 2015 IEEE 15th International Conference on, IEEE, 2015, pp. 1175–1180.

[15] R. Salakhutdinov, J. B. Tenenbaum, A. Torralba, Learning with hierarchical-deep models, IEEE transactions on pattern analysis and machine intelligence 35 (8) (2013) 1958–1971.

[16] M. R. Malgireddy, I. Nwogu, V. Govindaraju, Language-motivated approaches to action recognition, The Journal of Machine Learning Research 14 (1) (2013) 2189–2212.

[17] T. Hospedales, S. Gong, T. Xiang, A markov clustering topic model for mining behaviour in video, in: Computer Vision, 2009 IEEE 12th International Conference on, IEEE, 2009, pp. 1165–1172.

[18] M. D. Hoffman, D. M. Blei, C. Wang, J. Paisley, Stochastic variational inference, The Journal of Machine Learning Research 14 (1) (2013) 1303–1347.

[19] L. Deng, A tutorial survey of architectures, algorithms, and applications for deep learning, APSIPA Transactions on Signal and Information Processing 3.

[20] Y. Bengio, et al., Learning deep architectures for ai, Foundations and trends® in Machine Learning 2 (1) (2009) 1–127.

[21] G. E. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets, Neural computation 18 (7) (2006) 1527–1554.

[22] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, Journal of machine Learning research 3 (Jan) (2003) 993–1022.

[23] C. Fischer, Feedback on household electricity consumption: a tool for saving energy?, Energy efficiency 1 (1) (2008) 79–104.

[24] J. Kelly, W. Knottenbelt, The uk-dale dataset, domestic appliance-level electricity demand and whole-house demand from five uk homes, Scientific data 2 (2015) 150007.

[25] A. Filip, Blued: A fully labeled public dataset for event-based non-intrusive load monitoring research, in: 2nd Workshop on Data Mining Applications in Sustainability (SustKDD), 2011, p. 2012.

[26] J. Z. Kolter, M. J. Johnson, Redd: A public data set for energy disaggregation research, 2011.

[27] S. Makonin, F. Popowich, L. Bartram, B. Gill, I. V. Bajic, Ampds: A public dataset for load disaggregation and eco-feedback research, in: Electrical Power & Energy Conference (EPEC), 2013 IEEE, IEEE, 2013, pp. 1–6.

[28] S. Makonin, B. Ellert, I. V. Bajić, F. Popowich, Electricity, water, and natural gas consumption of a residential house in canada from 2012 to 2014, Scientific data 3.

[29] P. Davies, J. Dennis, J. Hansom, W. Martin, A. Stankevicius, L. Ward, Deep neural networks for appliance transient classification, in: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 8320–8324.

[30] C. Zhang, M. Zhong, Z. Wang, N. Goddard, C. Sutton, Sequence-to-point learning with neural networks for non-intrusive load monitoring, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[31] Z. Zheng, H. Chen, X. Luo, A supervised event-based non-intrusive load monitoring for non-linear appliances, Sustainability 10 (4) (2018) 1001.

[32] I. Abubakar, S. Khalid, M. Mustafa, H. Shareef, M. Mustapha, Application of load monitoring in appliances energy management–a review, Renewable and Sustainable Energy Reviews 67 (2017) 235–245.

[33] H. Kim, M. Marwah, M. Arlitt, G. Lyon, J. Han, Unsupervised disaggregation of low frequency power measurements, in: Proceedings of the 2011 SIAM International Conference on Data Mining, SIAM, 2011, pp. 747–758.

[34] J. Z. Kolter, T. Jaakkola, Approximate inference in additive factorial hmms with application to energy disaggregation, in: Artificial Intelligence and Statistics, 2012, pp. 1472–1482.

[35] M. J. Johnson, A. S. Willsky, Bayesian nonparametric hidden semi-markov models, Journal of Machine Learning Research 14 (Feb) (2013) 673–701.

[36] T. Y. Ji, L. Liu, T. S. Wang, W. B. Lin, M. S. Li, Q. H. Wu, Non-intrusive load monitoring using additive factorial approximate maximum a posteriori based on iterative fuzzy c-means, IEEE Transactions on Smart Grid (2019) 1–1doi:10.1109/TSG.2019.2909931.

[37] M. Wytock, J. Z. Kolter, Contextually supervised source separation with application to energy disaggregation, in: Twenty-eighth AAAI conference on artificial intelligence, 2014.

[38] M. Saad, B. Abdelhamid, Online gaussian lda for unsupervised pattern mining from utility usage data, submitted to ECML-PKDD.

[39] S. Mohamad, D. Arifoglu, C. Mansouri, A. Bouchachia, Deep online hierarchical unsupervised learning for pattern mining from utility usage data, in: UK Workshop on Computational Intelligence, Springer, 2018, pp. 276–290.

[40] M. J. Wainwright, M. I. Jordan, Graphical models, exponential families, and variational inference, Foundations and Trends® in Machine Learning 1 (1-2) (2008) 1–305.

[41] H. Robbins, S. Monro, A stochastic approximation method, The annals of mathematical statistics (1951) 400–407.

[42] M. Johnson, A. Willsky, Stochastic variational inference for bayesian time series models, in: International Conference on Machine Learning, 2014, pp. 1854–1862.

[43] M. Hoffman, F. R. Bach, D. M. Blei, Online learning for latent dirichlet allocation, in: advances in neural information processing systems, 2010, pp. 856–864.

[44] S. Mohamad, A. Bouchachia, M. Sayed-Mouchaweh, Asynchronous stochastic variational inference, arXiv preprint arXiv:1801.04289.

[45] S. Mohamad, M. Sayed-Mouchaweh, A. Bouchachia, Active learning for classifying data streams with unknown number of classes, Neural Networks 98 (2018) 1–15.

[46] S. Mohamad, A. Bouchachia, M. Sayed-Mouchaweh, A bi-criteria active learning algorithm for dynamic data streams, IEEE transactions on neural networks and learning systems 29 (1) (2016) 74–86.

Table. 7 Features after data pre-processing

TimestampNTP	Water	Electricity			HEMS						
		Real power	React. power	RMS spectr.	Gas	Temperature			Humidity	Radiator valve	Boiler firng
						Rooms	Radiators	Water			
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

6. Features extraction and DBN

For GLDA, we also drop DBN and replace the deep learning features extraction with NILM features known to work well with NILM supervised learning. Two types of features were used: low and high sampling rate features. The first type aims at capturing low sampling rate information such as the steady state of the appliance for example a change of steady-state active power measurement from a high to low value can identify whether the appliance is being turned On or Off. Active and Reactive power are among the most useful steady-state features [6]. The second type of features aims to capture transient behaviour between steady states, e.g., high sampling frequency of voltage noise (that results from the operation state change of the appliance). We consider the RMS spectrum power which involves (explicitly or implicitly) the information extracted with most transient-state features. Table 7 shows the obtained features over time windows of 1 second.

For DBN-LDA and DBN-LDA-HMM, deep learning is used for feature extraction. Here, the aligned data samples are the input of Deep Belief Network [21]. Pushed by the complexity of the mining task and motivated by the informativeness and simplicity of the water and sensors data, at this stage, we apply DBN only on the electricity data over time windows of 1 second.

The employed DBN [2] consists of three Restricted Boltzmann Machine layers where the first layer reduces the input dimension from 204911 (1 second granularity) to 700. The second and third layers' outputs dimensions are 200 and 100 respectively. Note that the first layer's inputs are from continuous space while the rest is categorical data. The rest parameters are left to the default setting.

[2] https://github.com/lmjohns3/py-rbm

The last layer's outputs are aligned and concatenated with the other utility and sensors discretised data.