State Space Model for the Prediction of Energy Consumption

Dexiong Chen Chia-Man Hung

March 1, 2017

Abstract

The data challenge proposed by Oze Energies aims at introducing new statistical models to predict future energy consumption, which is essential to obtain optimized procedures to simultaneously reduce costs and limit greenhouse gas emissions. Various models and methods have been discussed in the literature. We chose among them a linear state-space model with hidden states, and more precisely a Kalman filter and smoother, in combination with the Expectation-Maximization algorithm to estimate the parameters. We derived variants of strategies to adapt to our specific use case. By applying these methods to real-world data, validating on test dataset and visualizing the curves, we justified the robustness of our choice.

1 Introduction

In Europe, buildings account for 40% of total energy use and 36% of total CO_2 emission according to [3]. The prediction of energy use in buildings is therefore necessary to improve the utilization rate of energy, with the aim of achieving energy conservation and reducing anti-environmental impact. However, the energy system in buildings is complex, as the energy types and building types vary one with another. In the previous surveys, the main energy forms considered are heating/cooling load, hot water and electricity consumption. The most frequently considered building types are office, residential and engineering buildings, varying from small rooms up to big estates. In view of different geographical and environmental conditions, the energy behavior of a building can be influenced by many factors, such as weather conditions, especially the dry-bulb temperature, the building construction and thermal property of the physical materials used, the occupancy and their behavior, sub-level components such as lighting, HVAC (Heating, Ventilating, and Air-Conditioning) systems, their performance and schedules. All these factors lead the prediction of energy consumption for long term to a very difficult problem.

In recent years, various numerical models for describing thermal characteristics of building components have been investigated and developed. These models are extensively used in prediction and optimization of building energy consumption. For instance, [4, 10] survey diverse models and methods, by comparing explicitly their advantages and insufficiencies. However, the majority of these models provide only a short-term vision and low robustness. The efficiency of models in terms of forecasting is relatively adequate in some scenarios, but could be largely improved in certain other situations. Among all the related work, Thomas Berthou's thesis [1] has offered a comprehensive and systematic comparison between various models. It turns out that the state-space model shows a potential and competitive performance. The main advantages of this model is its tolerance with noise and well studied theoretical guarantees, such as confidence intervals.

In this report, we will investigate a linear dynamic model arising from the state-space model, present the theoretical results with time series analysis tools and study its prediction capacity and robustness in terms of building energy consumption. Our contributions are two folds. On one hand, we have implemented an expectation-maximization algorithm for the parameters estimation of the state space model. On the other hand, we have derived a variant model with is more adaptive for seasonal or periodic data, such as energy consumption in buildings. Thus, the report will be organized as follows. We begin with presenting the generic dynamic model that will be used to forecast the energy consumption. Then, we concentrate on the theoretical tools in order to estimate the parameters in the model. Based on the theoretical results, we apply the original method to the problem of the prediction of energy consumption, as well as a variant model which turns out to be more adaptive to the problem by taking the periodicity information into account. We compare the prediction performance and robustness for each model.

2 Problem and Models

The energy consumption is represented by the observations received from sensors in buildings characterizing thermal behaviors, in the form of a time series. We refer to these observations as $Y_1, Y_2, \dots, Y_n$ in $\mathbb{R}_+^d$, possibly multivariate random variables. We have also access to some exogenous data, which describes weather conditions or other environmental factors. We refer to these observations as $U_1, \dots, U_n$ in $\mathbb{R}^p$. The objective is to forecast the future energy consumption $Y_{n+1}, \dots, Y_{n+N}$ for some large integer N, given the historical thermal observations $Y_1, \dots, Y_n$ and the whole exogenous observations $U_1, \dots, U_N$. A natural and simple approach is to reformulate the problem as a regression problem

$$Y_{t+1} = f(Y_t, \dots, Y_{t-k}, U_{t+1}, \dots, U_{t+1-k'}), \tag{1}$$

where we try to express the energy use Y_{t+1} at time $t+1$ as a function of some fixed length of historical data $Y_{t-k}, \dots, Y_t$ and $U_{t+1-k'}, \dots, U_{t+1}$. Generally this model performs well for short-term prediction, while it suffers accumulated prediction errors and noise for long-term prediction as it uses predicted $\hat{Y}_{t+1}$ to continue predicting energy use at further time steps. For instance, if one uses a deep network to learn function f, the model may easily be overfitted and conduct to unfavorable prediction. However, if the model manages to characterize the variation of dynamics between time steps as well as resist to noise, then it can make the prediction of the denoising data instead of the noisy one.

Based on this notification, we consider the linear Gaussian state space model, which takes into account different types of noise within the data, observational noise and dynamic noise. Different from the dynamic noise, which enter the dynamics of the process, observational noise appears to the measurements as an additive effect. The state space model characterize the observations via a hidden chain and consists of a state equation describing dynamics of a process, and an observation equation modeling the measurement phase.

2.1 Linear State Space Model

The general linear Gaussian state space model can be represented as follows, with two equations as we explained above,

$$\begin{cases} X_{t+1} = A_t X_t + B_t U_t + \varepsilon_t \\ Y_t = C_t X_t + D_t U_t + \eta_t \end{cases} \tag{2}$$

where

- the error terms $\varepsilon_t \sim \mathcal{N}(0, P_t)$ and $\eta_t \sim \mathcal{N}(0, Q_t)$ are two independent vector-valued i.i.d. Gaussian sequences.
- X represents state vector sequence, which is not observable.
- X_0 is assumed to be $\mathcal{N}(\mu_0, \Sigma_0)$, which is independent of ε_t and η_t.
- $\theta_t = (A_t, B_t, C_t, D_t, P_t, Q_t, \mu_0, \Sigma_0)$ are parameters of the model.

As all variables are Gaussian, an obvious but useful remark is that all joint or conditional distributions of this model are also Gaussian. Thus they are fully determined by their mean vector and covariance matrix.

In order to estimate the parameters of this model, one can maximize the log-likelihood of the observations Y_t using unconstrained optimization methods such BFGS [6]. However, the expression of the log-likelihood is difficult to compute as well as its derivative. Noticing that the complete log-likelihood is much easier to compute, we consider another method called Expectation-Maximization (EM) algorithm.

2.2 Expectation Maximization Algorithm

EM algorithm is used to solve the general optimization problem, especially the equations cannot be resolved directly. In this paragraph, we will present the problem from a general point of view, and then applied EM algorithm to the state-space model.

2.2.1 Problem statement

we use here the definition described in [2]. Given a σ-finite measure λ on (X, χ), we consider a family $\{f(\cdot;\theta)\}_{\theta\in\Theta}$ of non- negative λ-integrable functions on X. This family is indexed by a parameter $\theta \in \Theta$, where Θ is a subset of $\mathbb{R}^{d_\theta}$ (for some integer d_θ). The task under consideration is the maximization of the integral

$$L(\theta) = \int f(x;\theta)\,\lambda(dx) \tag{3}$$

with respect to the parameter θ. $f(\cdot;\theta)$ might be thought as unnormalized probability density with respect to the measure λ. Thus $L(\theta)$ is the normalizing constant for $f(\cdot;\theta)$. $f(\cdot;\theta)$ is relatively a simple function on θ while $L(\theta)$ usually involves high dimensional integration and is therefore sufficiently complex to prevent the use of simple maximization approaches. we remark that it is not required that $L(\theta)$ be a likelihood, as any function satisfying 3 is a valid candidate.

In the following, we limit to a statistical setting. $f(\cdot;\theta)$ is thus associated to a density function $p(\cdot;\theta)$ defined by $p(\cdot;\theta) = f(\cdot;\theta)/L(\theta)$, and $L(\theta)$ is positive. maximizing $L(\theta)$ is equivalent to maximizing the *log-likelihood* $\ell(\theta) = \log L(\theta)$.

2.2.2 EM algorithm

The main ingredient of the EM algorithm is an auxiliary function known as the intermediate quantity. It is the family $\{\mathcal{Q}(\cdot;\theta')\}_{\theta'\in\Theta}$ of real-valued function on Θ, defined by

$$\mathcal{Q}(\theta;\theta') = \int \log f(x;\theta) p(x;\theta')\,\lambda(dx) \tag{4}$$

This term can be interpreted as the expectation of the function $\log f(X;\theta)$ for X distributed according to the probability density $p(x;\theta')$ with respect to the parameter θ'. By a simple calculation, $\mathcal{Q}(\theta;\theta')$ can be rewritten as

$$\mathcal{Q}(\theta;\theta') = \ell(\theta) - \mathcal{H}(\theta;\theta'), \tag{5}$$

where

$$\mathcal{H}(\theta;\theta') = -\int \log p(x;\theta) p(x;\theta')\,\lambda(dx) \tag{6}$$

With some hypotheses on regularity and integrability, we can get the essential result that justifies the correctness of the EM algorithm.

Theorem 2.1. *Under some assumptions, for any* $(\theta,\theta')\in\Theta^2$*, we have*

$$\ell(\theta) - \ell(\theta') \geq \mathcal{Q}(\theta;\theta') - \mathcal{Q}(\theta';\theta') \tag{7}$$

EM algorithm seeks to maximize $\ell(\theta)$ iteratively building a sequence $\{\theta^i\}_{i\geq 1}$ of parameter estimates given an initial guess θ^0. Each iteration is broken into two steps

- Expectation step (E-step): Determine $\mathcal{Q}(\theta;\theta')$.
- Maximization step (M-step): Choose θ^{i+1} that maximizes the intermediate quantity $\theta^{i+1} = \arg\max_{\theta\in\Theta} Q(\theta;\theta^i)$.

The correctness of EM algorithm is ensured by 7. To be precise, choosing θ to improve $Q(\theta;\theta^i)$ beyond $Q(\theta^i;\theta^i)$ can also improve $\ell(\theta)$ beyond $\ell(\theta^i)$ of at least the same quantity.

Now that the M-step is simply the computation of the derivative of Q with respect to θ, the main difficulty is the E-step, consisting of evaluating conditional expectations, which is radically an inference problem. To achieve this, we consider to use Kalman filter and smoother, a powerful tool in control theory used to estimate unknown variables for noisy ones. It can also be used in our situation, as to estimate hidden variables X_t in 2.

2.3 Kalman Filter and Smoother

Kalman filter is an algorithm to estimate recursively the hidden states using a series of noisy measurements observed in the past time. In contrast of Kalman filtering, Kalman smoothing is a backward pass which calculates a better estimate of states knowing all the measurements between the observations $Y_1,\ldots,Y_n$. More specifically, the estimations of the state vectors X_t are carried out by performing two passes through the data:

1. a forward pass, from $t = 0,\ldots,n$, using a recursive algorithm known as the Kalman filter that is applied to the observed time series;
2. a backward pass from $t = n,\ldots,0$ using recursive algorithms known as Kalman smoothers that are applied to the output of the Kalman filter.

2.3.1 Kalman Filter

The idea of Kalman filter is to update our knowledge of the system each time a new observation Y_t is brought in. It can be divided into two steps: *prediction* and *innovation*. These two steps calculate respectively an estimate of the state vector based on only the past observations (predicted state estimate) and that based on all past observations and the current observations (filtered state estimate).

In our treatment below, we define $Y_{0:t} = (Y_0, \ldots, Y_t)$. We denote $\hat{X}_{t|t} := \mathbb{E}[X_t|Y_{0:t}]$ the filtered state estimate X_t, $\hat{X}_{t|t-1} := \mathbb{E}[X_t|Y_{0:t-1}]$ the predicted state estimate of X_t and their respective associated variances $\Sigma_{t|t} := \mathbb{E}[(X_t - \hat{X}_{t|t})(X_t - \hat{X}_{t|t})^T] = \mathrm{Cov}(X_t - \hat{X}_{t|t})$, $\Sigma_{t|t-1} := \mathbb{E}[(X_t - \hat{X}_{t|t-1})(X_t - \hat{X}_{t|t-1})^T] = \mathrm{Cov}(X_t - \hat{X}_{t|t-1})$ to simplify our notation.

The idea of Kalman filter is to calculate $\hat{X}_{t|t}$ and $\Sigma_{t|t}$ by using $\hat{X}_{t-1|t-1}$ and $\Sigma_{t-1|t-1}$ when a new Y_t is bought in. We first calculate $\hat{X}_{t|t-1}$ and $\Sigma_{t|t-1}$, which is known as the prediction step.

$$\hat{X}_{t|t-1} = A_t \hat{X}_{t-1|t-1} + B_t U_t \tag{8}$$

$$\Sigma_{t|t-1} = A_t \Sigma_{t-1|t-1} A_t^T + P_t \tag{9}$$

To calculate $\hat{X}_{t|t}$ and $\Sigma_{t|t}$, we introduce first an important proposition in Bayesian regression theory.

Proposition 2.1 (Conditioning in the Gaussian Linear Model). *Let X and V be two independent Gaussian random vectors with $\mathbb{E}[X] = \mu_X$, $\mathrm{Cov}(X) = \sum_X$, and $\mathrm{Cov}(V) = \Sigma_V$, and assume $\mathbb{E}[V] = 0$. Consider the model*

$$Y = BX + V \tag{10}$$

where B is a deterministic matrix of appropriate dimensions. Further assume that $B\Sigma_X B^T + \Sigma_V$ is a full rank matrix. Then

$$\begin{aligned} \mathbb{E}[X|Y] &= \mathbb{E}[X] + \mathrm{Cov}(X,Y)\{\mathrm{Cov}(Y)\}^{-1}(Y - \mathbb{E}[Y]) & (11) \\ &= \mu_X + \Sigma_X B^t \{B\Sigma_X B^T + \Sigma_V\}^{-1}(Y - B\mu_X) & (12) \end{aligned}$$

and

$$\begin{aligned} \mathrm{Cov}(X|Y) &= \mathrm{Cov}(X - \mathbb{E}[X|Y]) = \mathbb{E}[(X - \mathbb{E}[X|Y])X^T] & (13) \\ &= \Sigma_X - \Sigma_X B^T \{B\Sigma_X B^T + \Sigma_V\}^{-1} B\Sigma_X & (14) \end{aligned}$$

From this proposition and the fact that the predictor-to-filter update is obtained by computing the posterior distribution given $Y_{0:t}$ in the equivalent pseudo-model $X_t \sim \mathcal{N}(\hat{X}_{t|t-1}, \Sigma_{t|t-1})$ and $Y_t = C_t X_t + D_t U_t + \eta_t$, where η_t is $\mathcal{N}(0, Q_t)$ distributed and independent of X_t. In consequence, we have the filtering formula

$$\hat{X}_{t|t} = \hat{X}_{t|t-1} + \Sigma_{t|t-1} C_t^T (C_t \Sigma_{t|t-1} C_t^T + Q_t)^{-1} (Y_t - C_t \hat{X}_{t|t-1} - D_t U_t) \tag{15}$$

$$\Sigma_{t|t} = \Sigma_{t|t-1} - \Sigma_{t|t-1} C_t^T (C_t \Sigma_{t|t-1} C_t^T + Q_t)^{-1} C_t \Sigma_{t|t-1} \tag{16}$$

However, there exists a different approach called innovation approach. It is based on the projection theorem in a Hilbert space.

2.3.2 Kalman Smoother

Now we consider computing the smoothed state estimate $\hat{X}_{t|n} = \mathbb{E}[X_t|Y_{0:n}]$ and $\Sigma_{t|n} = \mathrm{Cov}[X_t - \hat{X}_{t|n}] = \mathbb{E}[(X_t - \hat{X}_{t|n})(X_t - \hat{X}_{t|n})^T]$ via a backward recursion. We notice by independence that $\mathbb{E}[X_t|Y_{0:n}] = \mathbb{E}[\mathbb{E}[X_t|Y_{0:n}, X_{t+1}]|Y_{0:n}] = \mathbb{E}[\mathbb{E}[X_t|Y_{0:t}, X_{t+1}]|Y_{0:n}]$, with $X_{t+1} = A_tX_t + B_tU_t + \varepsilon_t$ and proposition 2.1, we have

$$\mathbb{E}[X_t|Y_{0:t}, X_{t+1}] = \mathbb{E}[\mathbb{E}[X_t|Y_{0:t}]+\mathrm{Cov}(X_t, X_{t+1}|Y_{0:t})\,\mathrm{Cov}(X_{t+1}|Y_{0:t})^{-1}(X_{t+1}-\mathbb{E}(X_{t+1}|Y_{0:t}))|Y_{0:n}]$$

$$\mathbb{E}[X_t|Y_{0:t}, X_{t+1}] = \hat{X}_{t|t} + J_t(X_{t+1} - \hat{X}_{t+1|t}) \tag{17}$$

where $J_t = \Sigma_{t|t}A_t^T\Sigma_{t+1|t}^{-1}$.

$$\hat{X}_{t|n} = \hat{X}_{t|t} + J_t(\hat{X}_{t+1|n} - \hat{X}_{t+1|t}) \tag{18}$$

By analogy, we get the recursion for $\Sigma_{t|n}$:

$$\Sigma_{t|n} = \Sigma_{t|t} + J_t(\Sigma_{t+1|n} - \Sigma_{t+1|t})J_t^T \tag{19}$$

This algorithm gives us a iterative backward recursion to compute the smoothers by computing first the Kalman filter. It is used to update the state vectors in the EM algorithm in each iteration, which we will present in the next section.

We can also compute the lag-one covariance smoother $\Sigma_{t,t-1|n} := \mathbb{E}[(X_t-\hat{X}_{t|n})(X_{t-1}-\hat{X}_{t-1|n})^T] = \mathrm{Cov}[X_t - \hat{X}_{t|n}, X_{t-1} - \hat{X}_{t-1|n}]$ (defined in [9]), which will be used in EM algorithm, thanks to the equality (17):

$$\begin{aligned}
\Sigma_{t,t-1|n} &= \mathbb{E}[\mathbb{E}[(X_t - \hat{X}_{t|n})(X_{t-1} - \hat{X}_{t-1|n})^T|X_t, Y_{0:n}]] \\
&= \mathbb{E}[(X_t - \hat{X}_{t|n})(\mathbb{E}[X_{t-1}|X_t, Y_{0:t-1}] - \hat{X}_{t-1|n})^T] \\
&= \mathbb{E}[(X_t - \hat{X}_{t|n})(J_{t-1}X_t + \underbrace{\hat{X}_{t-1|t-1} - J_{t-1}\hat{X}_{t|t-1} - \hat{X}_{t-1|n}}_{\mathcal{F}_n^Y-\text{measurable}})^T] \\
&= \Sigma_{t|n}J_{t-1}^T
\end{aligned}$$

2.4 EM Algorithm with Kalman Smoother

In this section, we suppose the parameters are invariant with the time in our state-space model. The observed data $Y_{1:n} = (Y_1, \dots, Y_n)$ is a subset of the not fully observable *complete data* $(X_{0:n}, Y_{1:n})$, where $X_{0:n} = (X_0, \dots, X_n)$ are the unobserved states. We assume the the joint distribution of $X_{0:n}$ and $Y_{1:n}$, for a given parameter $\theta = (A, B, C, D, P, Q, \mu_0, \Sigma_0)$, has a probability density $f(X_{0:n}, Y_{1:n}; \theta)$ with respect to the product measure $\lambda_n \otimes \mu_n$, which is referred to as the complete data likelihood. The Likelihood of the observation data is obtained by marginalization as

$$L(Y_{1:n}; \theta) = \int f(x, Y_{1:n}; \theta)\, \lambda_n(dx) \tag{20}$$

the family of probability density functions $\{p(\cdot; \theta)\}$ can be interpreted as

$$p(X_{0:n}|Y_{1:n}; \theta) = \frac{f(X_{0:n}, Y_{1:n}; \theta)}{L(Y_{1:n}; \theta)} \tag{21}$$

By abuse of notation, we denote $P(\cdot;\theta)$ the probability density function of any variable, given the parameter θ. We can compute explicitly $P(X_{0:n}, Y_{1:n};\theta)$ in our state-space model, since the $Y_t|X_t$ are independent

$$\begin{aligned}\log P(X_{0:n}, Y_{1:n};\theta) &= \log P(Y_{1:n}|X_{0:n};\theta) + \log P(X_{0:n};\theta) \\ &= \log P(X_0;\theta) + \sum_{t=0}^{n} \log P(Y_t|X_t;\theta) + \sum_{t=1}^{n} \log P(X_t|X_{t-1};\theta)\end{aligned}$$

And we have $Y_t|X_t \sim \mathcal{N}(CX_t + DU_t, Q)$, then

$$\log P(Y_t|X_t;\theta) = -\frac{1}{2}\left[d_y \log(2\pi) + \log|Q| + (Y_t - CX_t - DU_t)^T Q^{-1}(Y_t - CX_t - DU_t)\right] \tag{22}$$

In addition, $X_t|X_{t-1} \sim \mathcal{N}(AX_{t-1} + BU_{t-1}, P)$, then

$$\log P(X_t|X_t{-}1;\theta) = -\frac{1}{2}\left[d_x \log(2\pi) + \log|P| + (X_t - AX_{t-1} - BU_t)^T P^{-1}(X_t - AX_{t-1} - BU_t)\right] \tag{23}$$

and

$$P(X_0;\theta) = -\frac{1}{2}\left[d_x \log(2\pi) + \log|\Sigma_0| + (X_0 - \mu_0)^T \Sigma_0^{-1}(X_0 - \mu_0)\right] \tag{24}$$

As result,

$$\begin{aligned}\log P(X_{0:n}, Y_{1:n};\theta) &= K - \frac{1}{2}[\log|\Sigma_0| + n\log|P| + (n+1)\log|Q| \\ &+ (X_0 - \mu_0)^T\Sigma_0^{-1}(X_0 - \mu_0) \\ &+ \sum_{t=1}^{n}(X_t - AX_{t-1} - BU_{t-1})^T P^{-1}(X_t - AX_{t-1} - BU_{t-1}) \\ &+ \sum_{t=0}^{n}(Y_t - CX_t - DU_t)^T Q^{-1}(Y_t - CX_t - DU_t)]\end{aligned}$$

where $K = -\frac{1}{2}[nd_y \log(2\pi) + (n+1)d_x \log(2\pi)]$. Then the intermediate quantity is given by

$$\mathcal{Q}(\theta;\theta^i) = \mathbb{E}[\log P(X_{0:n}, Y_{1:n};\theta)|Y_{1:n}, \theta^i] \tag{25}$$

We have finished the E-step until now. To compute θ^i, we need to find the maximum point for each parameter by deriving $\mathcal{Q}(\theta;\theta^i)$. And we found

$$\frac{\partial\mathcal{Q}}{\partial A} = -2P^{-1}\mathbb{E}\left[\sum_{t=1}^{n}(X_t - AX_{t-1} - BU_{t-1})X_{t-1}^T|Y_{1:n}\right]$$

$$\hat{A}^{i+1} = \left(\sum_{t=1}^{n}\mathbb{E}[X_tX_{t-1}^T|Y_{1:n}] - \hat{B}^iU_{t-1}\mathbb{E}[X_{t-1}|Y_{1:n}]^T\right)\left(\sum_{t=1}^{n}\mathbb{E}[X_{t-1}X_{t-1}^T|Y_{1:n}]\right)^{-1} \tag{26}$$

$$\frac{\partial\mathcal{Q}}{\partial B} = -2P^{-1}\mathbb{E}\left[\sum_{t=1}^{n}(X_t - AX_{t-1} - BU_{t-1})U_{t-1}^T|Y_{1:n}\right]$$

$$\hat{B}^{i+1} = \left(\sum_{t=1}^{n}(\mathbb{E}[X_t|Y_{1:n}] - \hat{A}^i\mathbb{E}[X_{t-1}|Y_{1:n}])U_{t-1}^T\right)\left(\sum_{t=1}^{n}U_{t-1}U_{t-1}^T\right)^{-1} \tag{27}$$

$$\begin{aligned}\hat{P}^{i+1} \quad = \quad & \frac{1}{n}\sum_{t=1}^{n}(\mathbb{E}[X_t|Y_{1:n}] - \hat{A}^i\mathbb{E}[X_{t-1}|Y_{1:n}] - \hat{B}^iU_{t-1})(\mathbb{E}[X_t|Y_{1:n}] - \hat{A}^i\mathbb{E}[X_{t-1}|Y_{1:n}] - \hat{B}^iU_{t-1})^T \\ & +\hat{A}^i \operatorname{Var}(X_t|Y_{1:n})(\hat{A}^i)^T + \operatorname{Var}(X_t|Y_{1:n}) \\ & - \operatorname{Cov}(X_t, X_{t-1}|Y_{1:n})(\hat{A}^i)^T - \hat{A}^i \operatorname{Cov}(X_{t-1}, X_t|Y_{1:n}) \end{aligned} \tag{28}$$

$$\hat{C}^{i+1} = \left(\sum_{t=0}^{n}(Y_t - \hat{D}^iU_t)\mathbb{E}[X_t|Y_{1:n}]^T\right)\left(\sum_{t=0}^{n}\mathbb{E}[X_tX_t^T|Y_{1:n}]\right)^{-1} \tag{29}$$

$$\hat{D}^{i+1} = \left(\sum_{t=0}^{n}(Y_t - \hat{C}^i\mathbb{E}[X_t|Y_{1:n}])U_t^T\right)\left(\sum_{t=0}^{n}U_tU_t^T\right)^{-1} \tag{30}$$

$$\hat{Q}^{i+1} = \frac{1}{n+1}\sum_{t=0}^{n}(Y_t-\hat{C}^i\mathbb{E}[X_t|Y_{1:n}]-\hat{D}^iU_t)(Y_t-\hat{C}^i\mathbb{E}[X_t|Y_{1:n}]-\hat{D}^iU_t)^T+\hat{C}^i \operatorname{Var}(X_t|Y_{1:n})(\hat{C}^i)^T \tag{31}$$

$$\hat{\mu}_0^{i+1} = \mathbb{E}[X_0|Y_{1:n}] \tag{32}$$

$$\hat{\Sigma}_0^{i+1} = \mathbb{E}[X_0X_0^T|Y_{1:n}] - \hat{\mu}_0^i\mathbb{E}[X_0|Y_{1:n}]^T - \mathbb{E}[X_0|Y_{1:n}](\hat{\mu}_0^i)^T + \hat{\mu}_0^i(\hat{\mu}_0^i)^T \tag{33}$$

We remark that

$$\begin{aligned}\mathbb{E}[X_tX_{t-1}^T|Y_{1:n}] \quad &= \quad \operatorname{Cov}(X_t, X_{t-1}|Y_{1:n}) + \mathbb{E}[X_t|Y_{1:n}]\mathbb{E}[X_{t-1}|Y_{1:n}]^T \\ \mathbb{E}[X_tX_t^T|Y_{1:n}] \quad &= \quad \operatorname{Var}(X_t|Y_{1:n}) + \mathbb{E}[X_t|Y_{1:n}]\mathbb{E}[X_t|Y_{1:n}]^T \\ \operatorname{Cov}(X_t, X_{t-1}|Y_{1:n}) \quad &= \quad \operatorname{Cov}(X_{t-1}, X_t|Y_{1:n})^T\end{aligned}$$

All the parameters could be totally determined if $\mathbb{E}[X_t|Y_{1:n}]$, $\operatorname{Cov}(X_t|Y_{1:n})$ and $\operatorname{Cov}(X_t, X_{t-1}|Y_{1:n})$ are determined. These values can be computed thanks to the Kalman smoothing.

3 Prediction of Energy Consumption

We have implemented implemented Kalman filter and smoother together with EM-algorithm in Python. The data is provided by Oze Energies and can be found at https://challengedata.ens.fr/en/challenge/18/oze_energies_optimizing_energy_consumptions.html.

The goal is to predict the consumptions measured by sensors in the building to heat and cool the air based on observations, such as internal temperature, outside temperature, humidity, and building occupancy. However, the input data (observations) are only given in an anonymous way, i.e. we do not know exactly which observation each variable correspond to. Same goes with the output data. To make it clear, we are given a training input file, a training output file, and a testing input file, like in most data challenges. The performance of our model is evaluated on our prediction of the testing input file, using a mean square error metric.

We first looked at our data and realized that there are missing values in the exogenous data U, which makes them hard to be taken into account. We chose to do some data preprocessing by applying a Kalman filter to estimate the missing values, rather than discarding all rows with missing values.

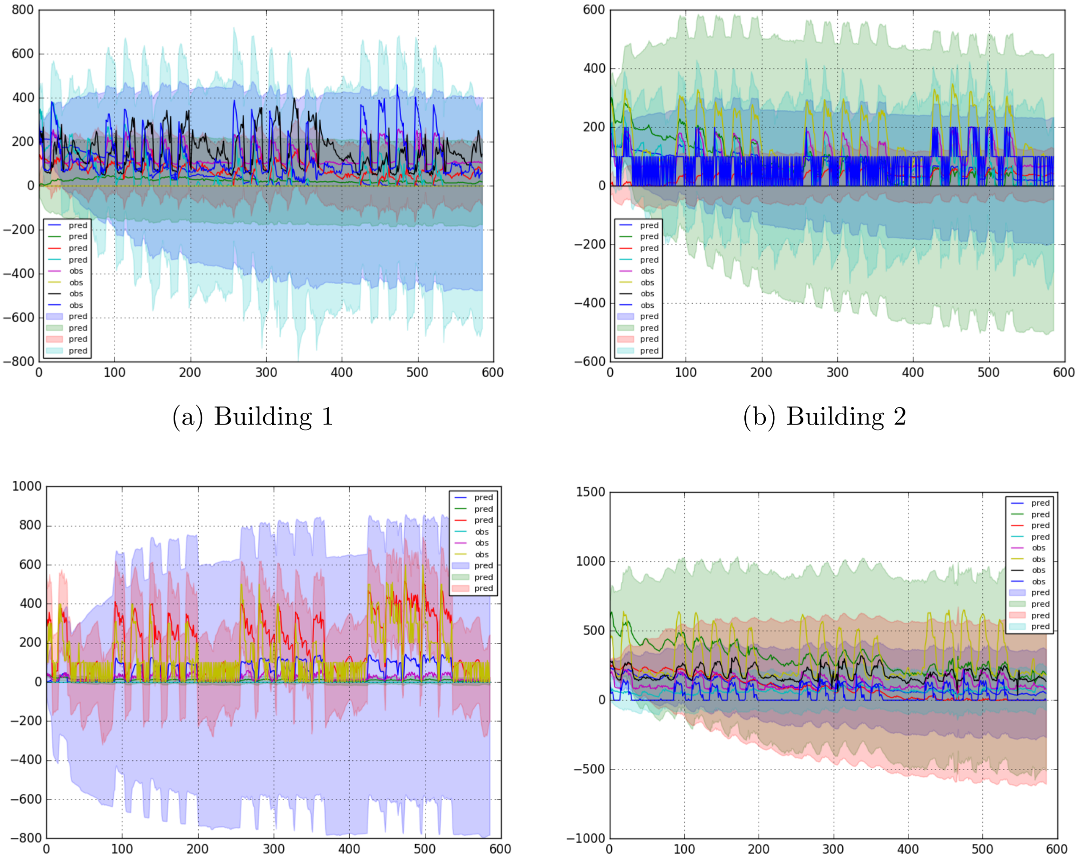
We propose here three different strategies based on the state space model, which reflects our track for the amelioration of the prediction performance.

Single-regime strategy. We apply the model described previously on the training data directly. In this strategy, we set $B = 0$ as it plays a similar role as D and we consider the entire time series as one single regime.

In fact, there are parameters that need to be decided. More precisely, we tune the dimension of the hidden state and number of iterations by splitting the training data into two parts, one for training and one for validation, to perform validation. The number of iterations does not have much impact if it converges. It turns out that setting the dimension of the hidden states to 2 makes it easier to converge and yields better results according to our experiments. Results are shown in Table 1 and Fig 1.

building	state dim	MSE
1	2	35693
2	2	22737
3	2	17371
4	2	48977

Table 1: Single-regime strategy: 20% of the training data used for validation, 500 iterations performed; Best results for each building.



(a) Building 1

(b) Building 2

(c) Building 3

(d) Building 4

Figure 1: Single-regime strategy: Prediction curves with confidence interval on part of the training data.

Multi-regime strategy. In this strategy, we take advantage of the periodicity within the data by dividing observations into different groups according to their regime. More precisely, we consider the following model

$$\begin{cases} X_{t+1} = A_{\sigma(t)} X_t + B_{\sigma(t)} U_t + \varepsilon \\ Y_t = C_{\eta(t)} X_t + D_{\eta(t)} U_t + \eta \end{cases} \tag{34}$$

where $\sigma : \mathbb{N} \to I$, $\eta : \mathbb{N} \to J$ and I, J are finite sets. Here I and J describe the set of different regimes. Regarding the parameter estimation, we use the EM algorithm to estimate the parameters respectively in each group.

Under this framework, the hidden states are naturally divided into five regimes: "night", "day", "day-to-night", "night-to-day" and "weekend". In practice, to achieve a good choice of regimes, we need to choose carefully the hours for the beginning of day or night, and also the length for the transition between two regimes. Note that the regimes can also be learned with the training data, by looking at the peaks and troughs. In this data challenge, we have fixed the regime period and chosen the most appropriate regime periods for each building.

Results are shown in Table 2 and Figure 2. This strategy significantly outperforms the single-regime strategy. Nevertheless, the method turns out to be unstable when inadequate regime parameters are used.

building	state dim	day start	day end	night start	night end	MSE
1	2	8	18	21	5	8130
2	2	10	18	23	6	5323
3	2	9	18	22	5	8825
4	2	9	18	22	5	6007

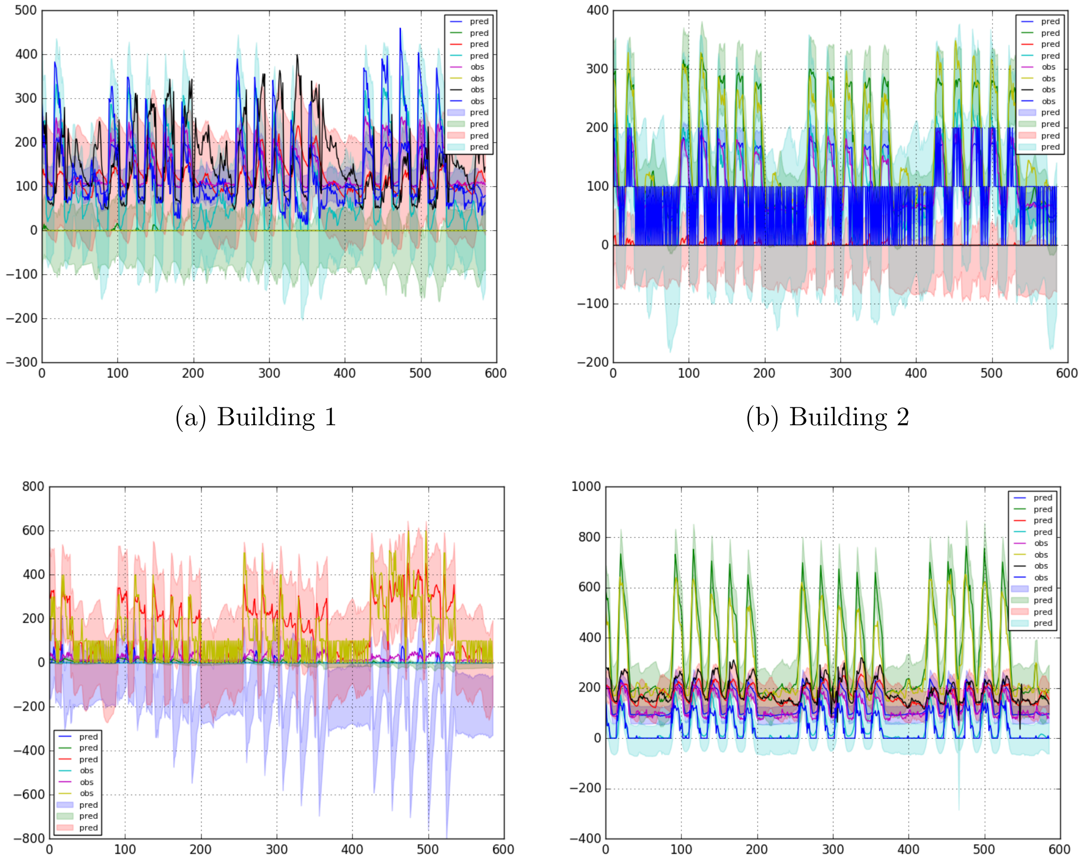
Table 2: Multi-regime strategy: 20% of the training data used for validation, 500 iterations performed; Best results for each building.

Two-lag multi-regime strategy. In this strategy, we use one more historical exogenous observation for the prediction. Specifically, we replace U_t by $\tilde{U_t} = (U_t, U_{t-1})$ in the above model 34, and use the same procedures for the prediction.

Results are shown in Table 3 and Figure 3. We notice that the prediction error is smaller than the multi-regime model.

building	state dim	day start	day end	night start	night end	MSE
1	2	8	18	21	5	8043
2	2	10	18	23	6	4568
3	2	8	18	22	5	6491
4	2	9	18	22	5	5032

Table 3: Two-lag multi-regime strategy: 20% of the training data used for validation, 500 iterations performed; Best results for each building.



(a) Building 1

(b) Building 2

(c) Building 3

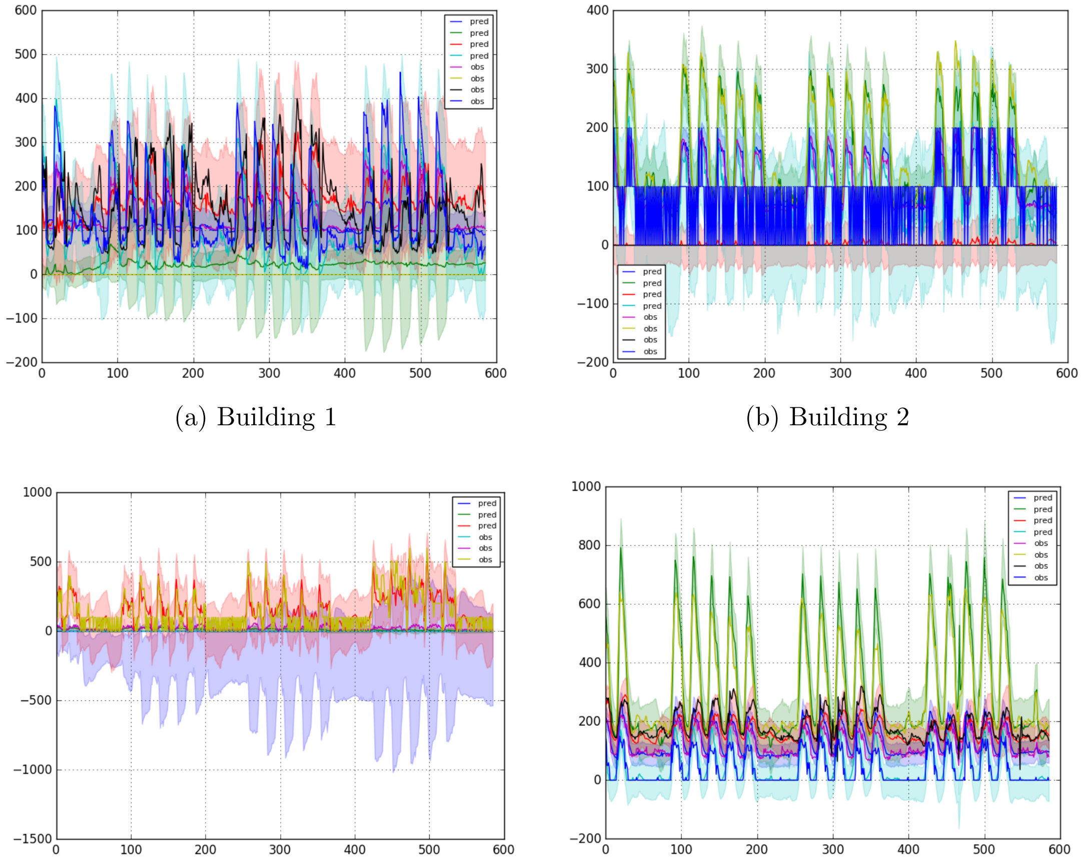
(d) Building 4

Figure 2: Multi-regime strategy: Prediction curves with confidence interval on part of the training data.

4 Discussions and Conclusions

We have successfully applied a linear state-space model for the prediction of energy consumption, by using the EM algorithm to estimate parameters. We have investigated and developed three strategies for the problem, by taking advantage of the specific invariance and periodicity of the data. According to the peaks and troughs in observations, we have divided observations into different groups, which leads to our multi-regime model. It turns out that the multi-regime model outperforms the original one, in terms of prediction error. With more historical exogenous observations, the two-lag multi-regime model gives even higher accuracy for the prediction.

As the linear state space model performs surprisingly well, some future work can also be considered. First, we can extend the linear model to a non-linear one, by using extended or unscented Kalman filter [5, 7, 8] and EM algorithm for parameter estimation. Another aspect that can be improved is the regime function σ and η, which we have fixed the regime periods in the project. However, an adaptive learning for the regime function can be considered to improve the performance. Besides, we have only evaluated the two-lag multi-regime model. More historical exogenous observations can be used in model for further comparisons. Furthermore, we have noticed that the prediction accuracy on the test data is not as good as that on the validation data.



(a) Building 1

(b) Building 2

(c) Building 3

(d) Building 4

Figure 3: Two-lag multi-regime strategy: Prediction curves with confidence interval on part of the training data.

One reason is that we have prefilled the missing values in the exogenous variables, which may lead to accumulated errors. Thus, a natural idea is to model the exogenous observations by another state space model, and estimate the parameters simultaneously for thermal and exogenous model.

References

[1] T. Berthou. *Development of building models for load curve forecast and design of energy optimization and load shedding strategies.* Theses, Ecole Nationale Supérieure des Mines de Paris, Dec. 2013.

[2] O. Cappé, E. Moulines, and T. Rydén. *Inference in hidden Markov models.* Springer Science & Business Media, 2006.

[3] D. EU. 31/eu on the energy performance of buildings. *European Paliament and Council, Brussels*, 2010.

[4] M. J. Jimenez and H. Madsen. Models for describing the thermal characteristics of building components. *Building and Environment*, 43(2):152–162, 2008.

[5] S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[6] M. F. Møller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural networks*, 6(4):525–533, 1993.

[7] M. I. Ribeiro. Kalman and extended kalman filters: Concept, derivation and properties. *Institute for Systems and Robotics*, 43, 2004.

[8] S. Roweis and Z. Ghahramani. Learning nonlinear dynamical systems using the expectation–maximization algorithm. *Kalman filtering and neural networks*, 6:175–220, 2001.

[9] R. H. Shumway and D. S. Stoffer. *Time series analysis and its applications: with R examples.* Springer Science & Business Media, 2010.

[10] H.-x. Zhao and F. Magoulès. A review on the prediction of building energy consumption. *Renewable and Sustainable Energy Reviews*, 16(6):3586–3592, 2012.