A Linear State-Space Model to Predict Hourly Electricity Demand

Renaud Alie[1] & Cédric Beaulac[2]

1 Department of Mathematics and Statistics, McGill University

2 Department of Statistical Sciences, University of Toronto

Introduction

In this project, we build a model that can instantaneously produce accurate estimates of the total hourly electricity demand given measurable environmental covariates.

Our model is based on the **linear State-Space** (LSS) paradigm. In this model, the latent state-space allows for a more expressive and flexible observed data distribution. The latent state is also at the center of the relationship between the response variable, the hourly electricity demand, and the environmental variable, the temperature, that serves as our predictor. The model is able to quickly fit large time series which is essential to utilize all the information contained in the hourly data set.

We established a list of expectations that we believe is important and want our prediction model to meet. These characteristics are central to most of our design decisions throughout this project. We want a model that:

- Produces predictions instantly; does not require re-training for every single prediction.
- Incorporates the information contained in the environment variables to improve the accuracy of the predictions.
- Allows us to gradually integrate more data; both vertically and horizontally.
- Allows us to slowly incorporate new components to the model in a successive manner.

Our assumptions

To construct a model that achieves the expectations we established, we made various assumptions about the prediction procedure. We assume that:

- Total hourly demand is the response variable.
- Environmental data is available up to current time during prediction.

Linear State-Space Model

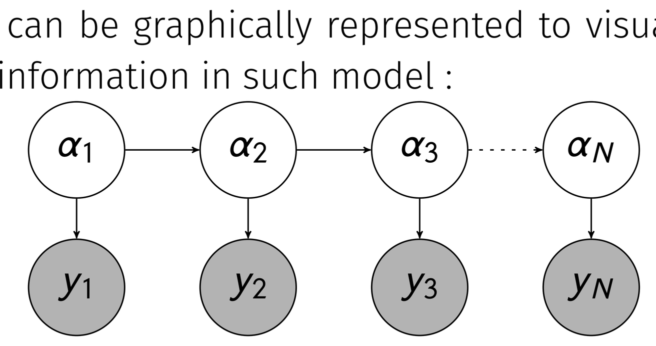
A linear State-Space model (LSSM) is a latent variable model. We defined α_t as the **unobserved latent state** at time t. The latent state α represents the traditional components of a time series such as trends and cycles. The set of α is perceived as an unobserved times series; they are temporally correlated in that α_t is a function of α_{t-1}. The various components of α are assumed to be Gaussian.

We define y as noisy observations of the unobserved time series. In a **linear** state-space model, y_t is a linear combination of α_t. Once again, variability on observations is accounted for with a Gaussian variable. Rigorously the model is defined with the following equations:

$$\begin{aligned} y_t &= Z\alpha_t + \varepsilon_t, & \varepsilon_t &\sim N(0, R), \\ \alpha_t &= \Phi\alpha_{t-1} + \eta_t, & \eta_t &\sim N(0, Q). \end{aligned} \quad (1)$$

with initial condition $\alpha_0 \sim N(\mu_0, \Sigma_0)$. Z and Φ are pre-determined matrix that simply encompass the linear combination.

A LSSM can be graphically represented to visualize the flow of information in such model :



The graphical model above assumes that the observations y are conditionally independent given the latent state α. The structure of dependence is the same as that of a hidden Markov model (HMM) but we make the additional hypothesis that the latent series and observed series are jointly Gaussian. This assumption is central to the latent space estimation technique and observation prediction technique, respectively the filtering algorithm and the forecasting algorithm.

Filtering

Filtering is the process of estimating the distribution of the latent state α_t given the observations

$$y_{1:t} := (y_1, y_2, \ldots, y_t)$$

up to time current time t. Under the Gaussian assumption, this can be done by evaluating the following conditional expectation and variance:

$$\begin{aligned} \hat{\alpha}_{t|t} &:= E[\alpha_t | y_{1:t}], \\ P_{t|t} &:= \mathrm{Var}(\alpha_t | y_{1:t}). \end{aligned}$$

Actually, filtering is a forward algorithm that recursively computes the filtered expectation and variance of the latent state, that is $\hat{\alpha}_{t|t}$ and $P_{t|t}$ for $t = 1, 2, \ldots, N$. Given initial values $\hat{\alpha}_{0|0} = \mu_0$ and $P_{0|0} = \Sigma_0$, the filtering algorithm successively computes $\hat{\alpha}_{t|t}$ and $P_{t|t}$ using the previous expectation $\hat{\alpha}_{t-1|t-1}$ and variance $P_{t-1|t-1}$ according to the model defined by (1).

Forecasting

Forecasting is the prediction process and a p step-ahead forecast for a LSSM can be performed by evaluating the following conditional expectation:

$$\hat{y}_{N+t|N} := E[y_{N+t} | y_{1:N}]$$

for $t = 1, 2, \ldots, p$ with N being the length of the observed series.

According to equations (1), the forecasted value $\hat{y}_{N+t|N}$ can be expressed using the conditional expectation of the latent state:

$$\hat{y}_{N+t|N} = ZE[\alpha_{N+t} | y_{1:N}]. \quad (2)$$

It can be shown in turn using equations (1) that the expected latent states admit the following recursion:

$$E[\alpha_{N+t} | y_{1:N}] = \Phi E[\alpha_{N+t-1} | y_{1:N}]. \quad (3)$$

Recall that the last expectation computed by the filtering algorithm is

$$\hat{\alpha}_N = E[\alpha_N | y_{1:N}].$$

When combining equations (2) and (3) with the N^{th} filtered state $\hat{\alpha}_N$, we obtain a simple way to compute the p step-ahead forecast:

$$\hat{y}_{N+t|N} = ZE[\alpha_{N+t} | y_{1:N}] = Z\Phi^t\hat{\alpha}_N,$$

for $t = 1, 2, \ldots, p$. This is considered the traditional forecasting technique from now on.

Multivariate LSSM

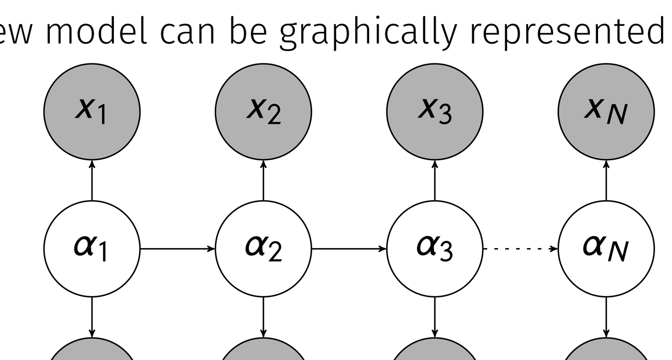
To incorporate an additional set of observed variables x, we define a new multivariate LSSM with two conditionally independent set of observed variables. We define:

$$x_t = \zeta\alpha_t + \rho_t, \qquad \rho_t \sim N(0, S),$$

The components of this multivariate LSSM are

- α_t: latent state
- y_t: response variables
- x_t: covariates

This new model can be graphically represented with :



Roughly speaking, this model assumes that x and y are two different noisy observations of the same underlying time series α. Our graphical representation suggests that y_t and x_t are conditionally independent given the state α_t.

The originality of this approach is that we do not consider the covariates as fixed parameters, but instead as another observed time series. Consequently, in the next section we describe a new forecasting technique that efficiently propagates the information from x to y.

Conditional Forecasting

Our main contribution is the adaptation of the traditional forecasting algorithm that we coined **conditional forecasting**. Our technique combines elements from both the filtering algorithm and the traditional forecasting algorithm.

Assume we observed both series x and y up to time N and we want to forecast p values of the response variables y. At time t, for $t \in (1, 2, \ldots, p)$ we observe predictors x from x_1 to x_{N+t} and try to estimate y_{N+t} with $\tilde{y}_{N+t|N}$. The conditional forecasting estimate can be stated as evaluating the following conditional expectation:

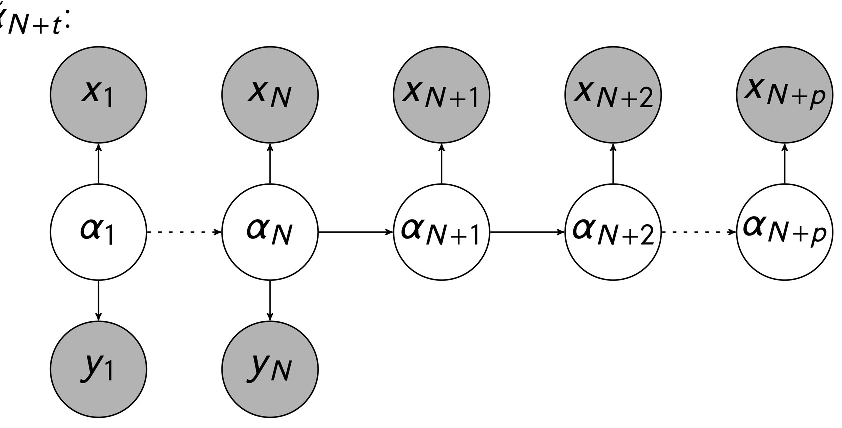
$$\tilde{y}_{N+t|N} := E[y_{N+t} | y_{1:N}, x_{1:N+t}]$$

for $t = 1, 2, \ldots, p$. Because of the conditional independence induced by our model, this forecasted value can be written in terms of the hidden state as

$$\tilde{y}_{N+t|N} = ZE[\alpha_{N+t} | y_{1:N}, x_{1:N+t}] := Z\tilde{\alpha}_{N+t}, \quad (4)$$

where Z is the pre-determined combination matrix from equations (1).

Finally, we adapted the filtering algorithm to run on the following network with provides us with the needed $\tilde{\alpha}_{N+t}$:



Consisting of a filtering part followed by a simple matrix multiplication our conditional forecasting algorithm is extremely fast (1000 predictions in 8.31 seconds).

In addition, if we receive new observations of the responses variables y, say p observations, we can run the traditional filtering algorithm from N to $N + p$ without having to restart from scratch which allows us to gradually integrate more data when the opportunity presents itself.

Implementation Choices

For this case study, we have set the hourly total electricity demand to be the response y. Our current implementation uses temperature as the environmental variable x, but our proposed formulation allows to incorporate more than one covariate.

In our current implementation, the latent state is a 32-dimensional variable composed of two main parts:

1. A (deterministic) cyclical component for both variables with daily, weekly and yearly periods encoded using linear combinations of sinusoidal functions.
2. A stochastic local-level component whose prior distribution is a Gaussian random walk with correlated steps between the response and the covariate.

Point #2 is key to the relation between x_t and y_t; the heuristic is that a large deviation of the temperature from its expected periodic components at time t should correspond to such a deviation of the total electricity demand at t. This is encoded in the local-level component of the latent state where we allowed the local-level of the predictor to be correlated with the local-level of the response.

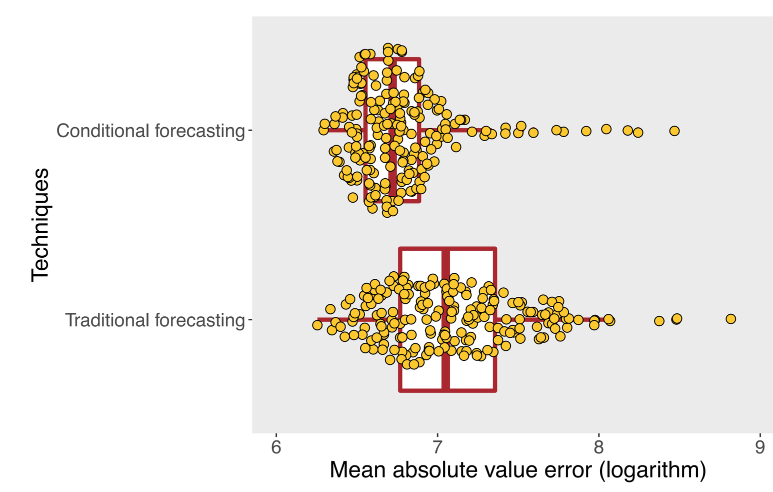
The magnitude of this correlation is established by cross-validation. We settled on a high correlation close to 1.

We also had to incorporate a dynamic pattern in the local-level component to model the fact that the electricity demand is negatively correlated with the temperature in the winter and positively instead in the summer.

Our implementation relies on clever uses of the efficient filtering algorithm provided with the `dlm` package in R and is available on GitHub at github.com/CedricBeaulac/SSC_CaseStudy_2020.

Experiments and Results

We now test our conditional forecasting algorithm. To begin, we train both models on three years of data. Then we predict the hourly electricity demand for two full months (1344 predictions) with traditional and conditional forecasting. Finally, we compute the error for both set of predictions; in this case we used the recommended mean absolute value error function. We repeated this procedure 200 times by randomly sampling a time point within the first 10 years of data. The results are shown bellow:



Empirically, the error of our forecasting technique has a lower expectation and has smaller variance than the error of the traditional forecasting. This improvement in accuracy leads us to believe that the model shows great potential for further improvement such as adding more environmental variables for instance. The plots bellow can help visualize the ability of our conditional forecasting algorithm to account for the deviation of temperature from its pure cyclical constituent during prediction:

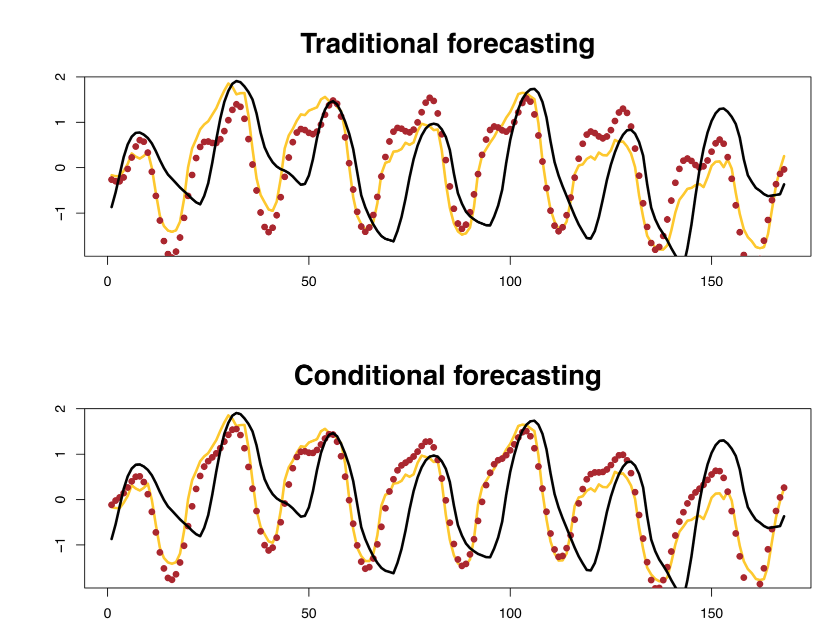


Figure: One week of hourly electricity demand (in yellow) and the associated predicted value (red dots) along with the temperature (black).

Takeaways

Strengths:

- Our proposed LSSM is very fast to train.
- Our conditional forecasting technique achieves the four characteristics we established in the introduction.
- The model is very flexible; we can include more components to the latent state.
- The model allows us to control the type of relationship between y and x by putting correlations only on certain components of the latent state (the local-level in this example).

Weaknesses:

- Our model contains many hyperparameters and cross-validation is difficult to perform.
- Our techniques are constructed upon a few heuristic decisions which could not be automated (no meta-learning possible at the moment).

References

[1] James Durbin and Siem Jan Koopman. *Time series analysis by state space methods.* Oxford University Press, 2012.

[2] Giovanni Petris. An R package for dynamic linear models. *Journal of Statistical Software*, 36(12):1–16, 2010.

[3] Robert H Shumway and David S Stoffer. *Time series analysis and its applications: with R examples.* Springer, 2017.