HIDDEN MARKOV MODELS
&
SEQUENTIAL MONTE CARLO METHODS

On particle filters applied to electricity load forecasting

Zakarya ALI
Antoine GRELETY
Samir TANFOUS

14 janvier 2018

1 Summary

"On particle filters applied to electricity load forecasting" is an article published in 2013 by Tristan Launay, Anne Philippe and Sophie Lamarche which relates recent developments regarding sequential Monte Carlo Methods and online prediction. They present, on the same issue, different use of particle filters : filtering, predicting, smoothing. The paper shows how they are a computationally cheaper approach than classic MCMC algorithms (with online prediction, the model does not need to be fully estimated every time a new observation is added), but also the inconvenients that must be taken care of to perform them. The application of these methods, with an implementable dynamic model, on electricity load forecasting are shown to be as effective as other proven methods. In our project, we decided to apply a revised version of the article dynamic model and assessed its efficiency in terms of prediction and estimated its parameters.

2 Model

We denote $\mathcal{N}(\mu, \Sigma)$ the Gaussian distribution with mean μ and variance Σ, and $\mathcal{N}(\mu, \Sigma, S)$ the corresponding truncated Gaussian distribution the support of which is S.
We want to simulate, for each day n, and each half hour i, the state $x_{n,i}$ such as :

$$y_{n,i} = x_{n,i} + v_{n,i},$$

where $y_{n,i}$ is the French electricity load at day n and half hour i and $v_{n,i} \sim \mathcal{N}(0, \sigma^2)$. For the rest of the document, we will not write the subscript i anymore.

The components of the model for a particular half hour are as follows :

$$x_n = x_n^{season} + x_n^{heat}$$

with,

$$\begin{cases} x_n^{season} = s_n.\kappa_{daytype_n} \\ x_n^{heat} = g_n^{heat}(T_n^{heat} - u^{heat})\mathbb{1}_{(u^{heat} > T_n^{heat})} \end{cases}$$

where,

$$s_n = s_{n-1} + \epsilon_n^s, \text{ with } \epsilon_n^s \sim \mathcal{N}(0, \sigma_{s,n}^2,]-s_{n-1}, +\infty[)$$

$$g_n^{heat} = g_{n-1}^{heat} + \epsilon_n^g, \text{ with } \epsilon_n^g \sim \mathcal{N}(0, \sigma_{g,n}^2,]-\infty, -g_{n-1}^{heat}[)$$

$$\sigma_{s,n} = \sigma_{s,n-1} + \eta_n^s, \text{ with } \eta_n^s \sim \mathcal{N}(0, \sigma_s^2,]-\sigma_{s,n-1}, +\infty[)$$

$$\sigma_{g,n} = \sigma_{g,n-1} + \eta_n^g, \text{ with } \eta_n^g \sim \mathcal{N}(0, \sigma_g^2,]-\sigma_{g,n-1}, +\infty[)$$

The fixed parameter vector is $\theta = (\sigma, \sigma_s, \sigma_g, u^{heat}, \kappa)$

By comparing to the article model, we did not use x^{cool} as we do not know how to retrieve the required data for it and, as the the paper points out, it has limited influence over French electricity load.

3 Data & Code

For French electricity load observations, y_n, we took RTE data which propose electricity consumption in France for every day, every 30 minutes. We used for our model the data from years 2015 to 2017.

For T_n^{heat}, we retrieved Meteo France every day temperatures (not a temperature for each 30 minutes but a temperature every 3 hours) with the same time window as electricity observations. We have only kept metropolitan values.

Regarding the code, see :
`https://github.com/zakaryaxali/smc_electricity_forecast/blob/master/README.md`

4 Particle Filter

4.1 Initialization

We applied a Gibbs sampler in order to initialize the set of parameters

$$(s_0, g_0^{heat}, \sigma_{s,0}^2, \sigma_{g,0}^2, \kappa, \sigma^2).$$

The prior densities of the parameters are sometimes very vague, especially the variances parameters defined by $\mathcal{IG}(10^{-2}, 10^{-2})$. The Particle Filter is therefore very sensitive to the initial parameters.
We ran the Gibbs sampler over the first 15 days of January 2016. As temperatures in January are low, this periods allows the heating effect (namely x^{heat}) to be non null so that we can observe and calibrate g_n^{heat} during the initialization phase.
The standard deviations simulated were not always realistic. Therefore we decided to set them by hand in an order of magnitude consistent with electricity load values. We set $\sigma_{g,0}^2$ and $\sigma_{s,0}^2$ at 10^8 and σ at 3.10^7.
We also realized that different values of κ set by hand do not improve significantly the predictions. In fact, the predictions are slightly more volatile than the case of κ invariant. We simplified the model so that κ is uniform with a value of 1 for each of the 9 day types. This implies that we assume no effect of weekends and bank holidays on electricity consumption.

4.2 Resampling

At each period n, we predict the electricity load and resample the weights if the degeneracy is too important. When the Effective Sample Size (ESS) drops below half the number of particules, we applied a multinomial resampling technique instead of the which is easier to implement than the residual resampling proposed in the article.
If the degeneracy is critical and drops below a threshold of $\epsilon = 10^{-3}$ multiplied by the number of particles, we do not update the weights at this period and move to the next step.
We observe that the weights degenerate below half the number of particles most of the time.

4.3 PMCMC

We ran a PMCMC algorithm using the particle filter described in the last paragraph. A calibration of the set of parameters $\Theta = (g_0^{heat}, \sigma^2_{s,0}, \sigma^2_{g,0}, \sigma^2)$ that best reflects the distribution of the real consumption motivates this idea.
At each iteration, PMCMC algorithm generates new proposal parameters and accepts them with acceptance probability r such that :

$$r = \frac{p(\theta^*)L(y \mid \theta^*)h(\theta_m \mid \theta^*)}{L(\theta_m)p(y \mid \theta_m)h(\theta^* \mid \theta_m)}$$

where :
-$L()$ is the likelihood estimated by the particle filter. At each iteration of the algorithm, we generate a new particle fitler with the new proposed parameters θ^*
-$p()$ is the joint prior distribution of the parameters. We used the same priors as the ones specified in the article : inverse gamma for the variances and standard normal for μ_{heat}.
-$h()$ is a gaussian independent proposal, therefore $h(\theta \mid \theta^*) = h(\theta)$

We actually applied the log-acceptance probability for computational reasons.

5 Results

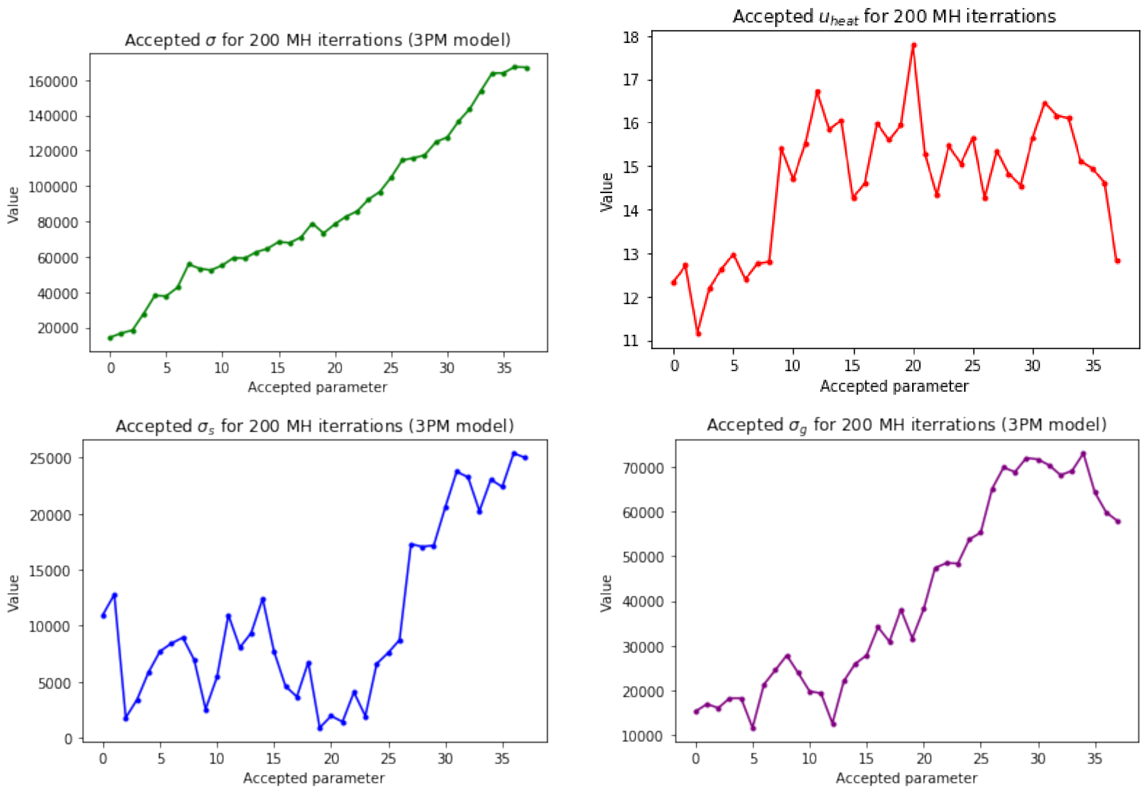
5.1 PMMH

We ran the PMMH algorithm over 200 iterations. At each iteration, the particle filter runs with the new proposed parameters over 715 days (nearly two years). It took about four hours to run the algorithm. We simulated PMMH at two different periods of the day : one at 3pm and another one at 3am. We chose initial values of 10^8 for the variances $(\sigma_s^2, \sigma_g^2, \sigma^2)$ and $\mu_{heat} = 13$. We also defined hyperparameters for the functions of the proposed parameters. Each of the truncated gaussian proposals of the variances take standard deviation of 5.10^3 and are centered around their current value. The proposal of μ_{heat} is a standard normal of mean its current value.

We observed that the log-likelihood took very large values negatively. This divergence could be caused for example by the initial calibration of the Particle Filter. As a result the other elements in the Metropolis Hastings proposal turned out to be negligible relatively to the log-likelihood estimated from the proposed parameters and the log-likelihood from the current parameters. The accept/reject decision is therefore essentially driven by the weight of the two log-likelihood estimations.

Acceptance rate	μ_{heat}	σ_s	σ_g	σ
29%	8.1	$1.8.10^4$	$8.4.10^4$	$2.9.10^5$

Table 1 : Estimated parameters PMMH 3am



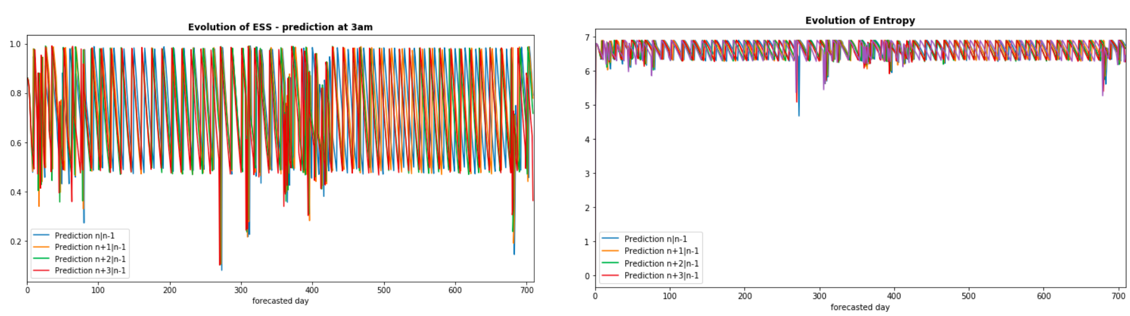
Acceptance rate	μ_{heat}	σ_s	σ_g	σ
19%	12.8	$5.7.10^4$	$2.5.10^4$	$1.6.10^5$

Table 2 : Estimated parameters PMMH 3pm

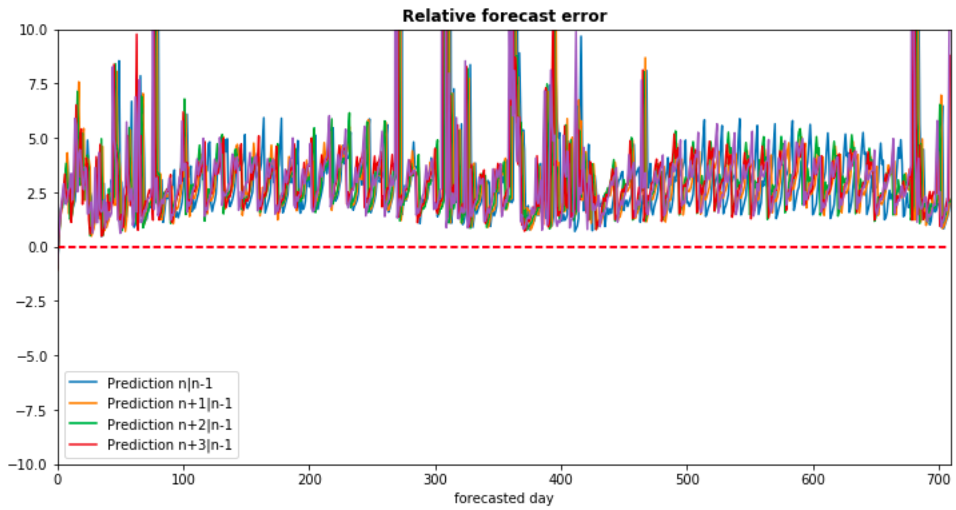
Table 1 and 2 above show the values of the parameters estimated by PMMH. In both cases, the value of σ is clearly overestimated according to the order of magnitude of the electricity load. σ_g and σ_s are also higher than expected.

5.2 Forecasts

We ran the particle filter with the new set of parameters simulated from PMMH to predict the electricity load from one to four days ahead for both 3am and 3pm.



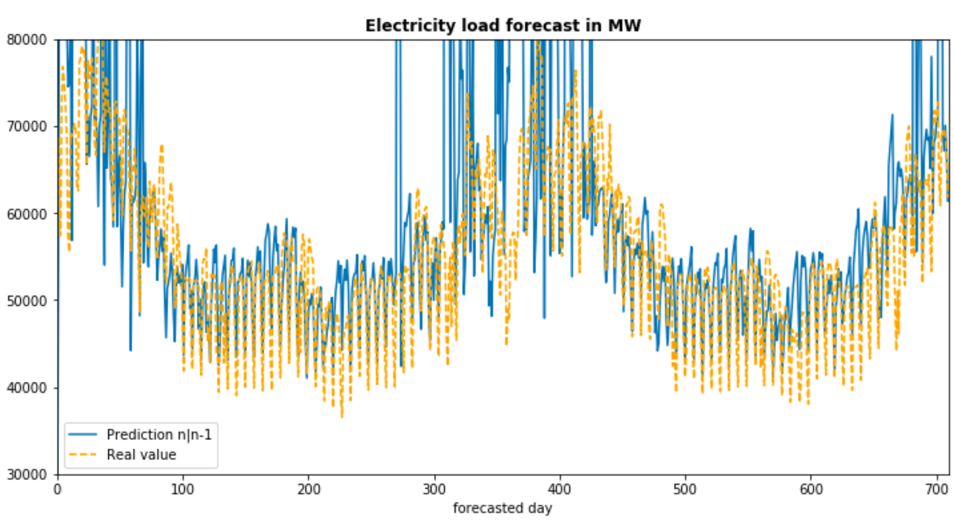
As we can see in the ESS graph above, the ESS and entropy values have very similar patterns for each forecast.



We have seen in the last section that the estimated variances were very large. The graph of the relative error above confirms our belief. The MAPE value is about 3.1. The predicted values are about 3 times larger than the true value in average.

5.3 Discussions

It is clear that the parameters have converged to higher values than expected. One of the reasons might be that the particle filter is highly sensitive to initial values. Starting from other initial values, the particle filter might have generated a likelihood less volatile and converged towards more reasonable values.
By means of more computational resources, we could have also used more particles at each iteration (10^5 particles in the article, only 10^3 particles in our case).



The graph above shows the electricity load prediction one day ahead over two years compared to the real value in MW, with the variances set by hand (using the values that we set at initialization phase). The MAPE value is about 0.19. The particle filter behaves much better with these values, except for a few outlying days such as during Chritmas period.

Conclusion

To conclude on our project, we tried to develop a Particle Filter very sensitive to initial values and computationally intensive. The results of our PMMH calibration cannot be used practically. However the particle filter predicts more reasonable values with parameters set by hand. Another way to estimate it is the Robbins-Monro algorithm. Although it is much simpler, we did not find the right parameter log-likelihood gradient to perform it.

During this project we have acknowledged the benefits of using a particle filter for online prediction and also the practical difficulties that come with, such as the calibration of initial values. As a line of thought, the parameter estimation we got could have been tested with a Kalman filter to compare both approaches and whether our θ was usable.