PhD Research Update

Sequential Bayesian Learning on State Space Models

Patrick Aschermayr

Supervisor: Dr. Kostas Kalogeropoulos

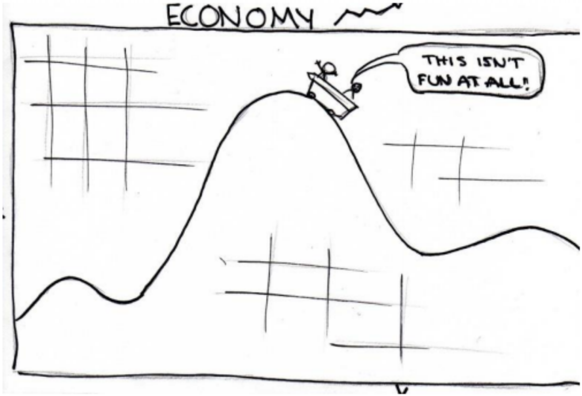
Co-Supervisor: Prof. Pauline Barrieu

June 17, 2022

Department of Statistics

Motivation

Economies move in cycles



Credit: *Economics fun*.

Inference

State Space Models

- A **State Space Model (SSM)** with parameter $\theta \in \mathbb{R}^D$ is a bivariate stochastic process $\{E_t, S_t\}_{t=1,2,\ldots}$, with the following distributional form:
  - $$\begin{aligned} \theta &\sim p(\theta), \\ S_0 &\sim p(s_0 \mid \theta), \\ S_t &\sim p(s_t \mid s_{0:t-1}, \theta), \\ E_t &\sim p(e_t \mid e_{1:t-1}, s_{0:t}, \theta). \end{aligned}$$
- The goal is to infer **the full posterior distribution** :

$$p(s_{0:T}, \theta \mid e_{1:T}) = \frac{p(e_{1:T} \mid s_{0:T}, \theta)\, p(s_{0:T} \mid \theta)\, p(\theta)}{p(e_{1:T})}. \tag{1}$$

- SSMs can handle **structural breaks**, shifts, or **time-varying parameters** of a model and still have an **interpretable structure**. They are **generative**, and allow for multi step forecasting, imputing missing data, and account for non-equal time steps.

Challenges

- **Marginal likelihood** $p(e_{1:T})$ intractable, but computation can be avoided.
- **Naive batch estimation** of high dimensional full posterior distribution $p(s_{0:T}, \theta \mid e_{1:T})$ computationally **unfeasible**.
- Need to efficiently **evaluate full posterior distribution iteratively** as $p(e_{1:T}, s_{0:T} \mid \theta) = p(s_0 \mid \theta) \prod_{t=1}^{T} p(s_t \mid s_{0:t-1}, \theta)\, p(e_t \mid e_{1:t-1}, s_{0:t}, \theta)$.
- **Marginal posterior distribution** $p(\theta \mid e_{1:T})$ difficult to compute, as $p(e_{1:t} \mid \theta) = \int p(e_{1:T}, s_{0:T} \mid \theta)\, ds_{0:T}$ is costly to evaluate or intractable.

Strategies and Examples

- If $\mathbf{S_{0:T}}$ **is continuous**, we can target $p(s_{0:T}, \theta \mid e_{1:T})$ via MCMC.
  - **Stochastic Volatility Model (SVM)**:
    - $e_t \sim N(\mu_0, e^{s_t})$,
    - $s_t \sim N(\mu + \phi(s_{t-1} - \mu), \sigma^2)$.
  - But how do we obtain state trajectory $p(s_{0:T} \mid \theta)$?
- If we can **integrate out** $\mathbf{S_{0:T}}$, can target $p(\theta \mid e_{1:T})$ directly.
  - **Hidden Markov Model (HMM)** :
    - $e_t \sim N(\mu_{s_t}, \sigma_{s_t})$,
    - $s_t \sim Categorical(p_{s_{t-1}})$.
  - Can compute $\sum_{s_{0:T}} p(e_{1:T}, s_{0:T} \mid \theta)$ in $\mathcal{O}(K^2 T)$ *.
  - What happens if s_t is non-Markovian?

(*) see (Baum and Petrie, 1966) . K = number of latent states, T = number of data points.

Strategies and Examples continued

- In HMM, $P(S_{t+k} = j, S_{t+1:t+k-1} = i \mid s_t = i)$ is implicitly geometric.
- **Hidden semi-Markov Models** * explicitly describe state durations:
  - $e_t \sim N(\mu_{s_t}, \sigma_{s_t})$
  - $s_t \sim \begin{cases} \delta(S_t = s_{t-1}) & d_{t-1} > 0, \text{ **} \\ P(S_t \mid s_{t-1}, d_{t-1}) & d_{t-1} = 0. \end{cases}$
  - $d_t \sim \begin{cases} \delta(S_t = s_{t-1}) & d_{t-1} > 0\ , \\ P(S_t \mid s_{t-1}, d_{t-1}) & d_{t-1} = 0\ . \end{cases}$
- Can compute $\sum_{s_{0:T}, d_{0:T}} p(e_{1:T}, s_{0:T}, d_{0:T} \mid \theta)$ in $\mathcal{O}(K^2(d_{max} - d_{min})^2 T)$ ***.

(*) see (Yu, 2010) and (Yu, 2016)

(**) $\delta(a, b)$ is the Kronecker product and equals 1 if $a = b$ and 0 otherwise.

(***) d_{min} = minimal state duration, d_{max} = maximal state duration, typically $(d_{max} - d_{min}) >> K$

Particle MCMC

- Independent of continuity of s_t, can **decompose problem** into targetting $p(s_{0:T} \mid e_{1:T}, \theta)$ and $p(\theta \mid e_{1:T}, s_{0:T})$:
  - Approximate $p(s_{0:T} \mid e_{1:T}, \theta)$ via a particle filter (PF *).
  - Target $p(\theta \mid e_{1:T}, s_{0:T})$ via MCMC.
  - Formally known as **Particle Gibbs** **.
- Can compute PF estimate for **both** $p(s_{0:T} \mid e_{1:T}, \theta)$ and $p(e_{0:T} \mid \theta)$ in $\mathcal{O}(NT)$ ***.

(*) see, e.g., Doucet and Johansen (2011)

(**) see (Andrieu and Roberts, 2009), (Andrieu et al., 2010), (Lindsten et al., 2014) and (Lindsten et al., 2015)

(***) N = number of particles, typically $K << N < T$.

Sequential Estimation

- Obtain **posterior predictive distribution** by integrating $s_{0:T}$ & θ:

$$
\begin{aligned}
p(e_{T+1} \mid e_{1:T}) &= \int p(e_{T+1}, s_{T+1}, s_{0:T}, \theta \mid e_{1:T})\, ds_{T+1}, s_{0:T}, \theta \\
&= \int p(e_{T+1} \mid s_{T+1}, s_{0:T}, \theta, e_{1:T})\, p(s_{T+1} \mid s_{0:T}, \theta, e_{1:T})\, p(s_{0:T}, \theta \mid e_{1:T})\, ds_{T+1}, s_{0:T}, \theta.
\end{aligned} \tag{2}
$$

- Sampling S_{T+1} and E_{T+1} trivial after $p(s_{0:T}, \theta \mid e_{1:T})$ is obtained.

- **Goal: sequentially explore** $p(s_{0:t}, \theta \mid e_{1:t})$ for $t = 1, \ldots, T$.

Sequential Monte Carlo Squared *

- Explore n sequences of distributions p($s^n_{0:t}, \theta^n \mid e_{1:t}$) for $t = 1, \ldots, T$.
  - Calculate $p(e_t \mid e_{1:t-1}, \theta^n)$, $p(e_{1:t} \mid \theta^n)$ and propagate $s^n_{0:t}$ **online** via PF.
  - If $p(e_t \mid e_{1:t-1}, \theta)$ estimates too noisy, jitter $s^n_{0:t}, \theta^n$ via Particle Gibbs.
  - ***Almost* real time**.
- Obtain **predictive distributions** for e_{t+1} and s_{t+1} and an estimate for **marginal likelihood** $p(e_{1:t})$ for each $t = 1, \ldots, T$.
- Use **CRPS** ** to compare predictive distribution of models. For forecasts $X_i, i = 1, \ldots, m$ and observation y, CRPS can be calculated as

$$CRPS(\hat{F}_m, y) = \frac{1}{m}\sum_{i=1}^{m}|X_i - y| - \frac{1}{2m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}|X_i - X_j|. \qquad (3)$$

(*) see Chopin (2002) and Chopin et al. (2013)

(**) see, e.g., (Jordan et al., 2019)

Applications

State Space Models and Financial Data

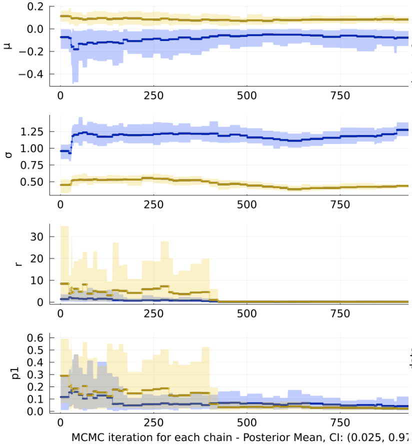
- **Stylized financial facts** *:
  - (u.1) Returns not iid, but show little serial correlation.
  - (u.2) Extreme returns appear in clusters.
  - (u.3) Returns have heavy tails.
  - (u.4) Volatility clusters and varies over time.
  - U.S. economic cycles widely vary in duration **.
- **Apply SMC^2 for HSMM on financial data**:
  - are model parameter constant across time?
  - how does HSMM fare against other SSM?

(*) see, e.g., (McNeil et al., 2005)

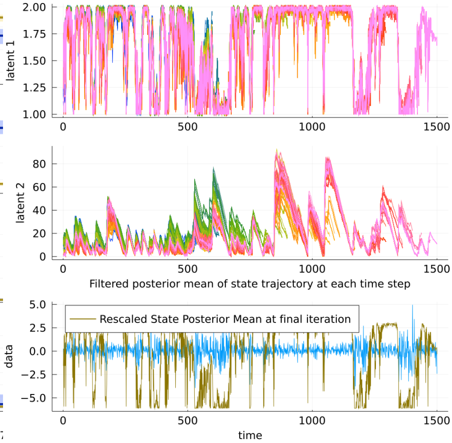
(**) see, e.g., https://www.nber.org/cycles.html

Results - HSMM

- **Traceplot for model parameter:**

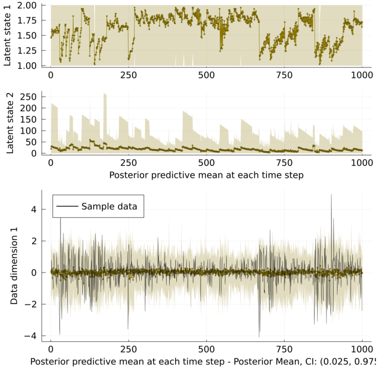


- **Traceplot for state trajectory:**



Results - Prediction and Model Comparison

- **SMC2 predictions:**



- **CRPS score for various models:**

Model	CRPS Score
HSMM	228.13
SV	229.43
HMM	230.15

  - More analysis needed!

Impact

Research contribution

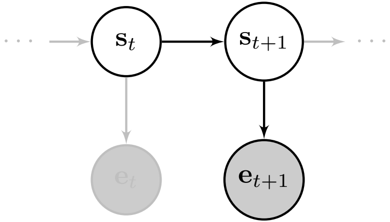
- **Model** and **Applications** contribution by
  - providing alternative ways for parameter estimation on HSSMs.
- **Algorithmic** contribution by
  - providing a toolbox for estimation and further inference on SSMs with arbitrary state and observation dependency that will be open sourced over the next months.
  - providing ideas for automatic adaption of SMC^2 tuning parameter, such as the number of jittering steps.

Discussion

Appendix

HMM as mixture distribution

HMM as sequential mixture:



$$
\begin{aligned}
P(e_{t+1} \mid s_t = k) &= \sum_{s_{t+1}} P(e_{t+1}, s_{t+1} \mid s_t = k) \\
&= \sum_{s_{t+1}} P(s_{t+1} \mid s_t = k) P(e_{t+1} \mid s_{t+1})
\end{aligned}
$$

Implicit geometric duration distribution of HMM

For a discrete 2-state, homogenous Markov chain, using the chain rule and the Markov assumption, it holds:

$$\begin{aligned}P(S_{t+3} = j, S_{t+2} = i, S_{t+1} = i \mid S_t = i) &= P(S_{t+3} = j \mid S_{t+2} = i)P(S_{t+2} = i, \mid S_{t+1} = i) \\ &= (1 - \mathcal{T}_{ii}) * \mathcal{T}_{ii}^2\end{aligned}$$

In general, for $t + k$ steps:

$$\begin{aligned}P(S_{t+k} = j, \ldots, S_{t+1} = i \mid S_t = i) &= (1 - \mathcal{T}_{ii}) * \mathcal{T}_{ii}^{k-1} \\ &= Geometric_{\mathcal{T}_{ii}},\end{aligned}$$

where the geometric distribution has to be interpreted as the length of state duration up to and including the transition to the other state.

HSMM distribution

In an EDHMM, transitions are allowed only at the end of each state, resulting in the following distributional forms:

$$S_t \mid s_{t-1}, d_{t-1} \sim P(S_t \mid s_{t-1}, d_{t-1}) = \begin{cases} \delta(S_t = s_{t-1}) & d_{t-1} > 0 \\ \mathcal{T}_{s_{t-1},.} & d_{t-1} = 0 \end{cases} \tag{4}$$

$$D_t \mid s_t, d_{t-1} \sim P(D_t \mid s_t, d_{t-1}) = \begin{cases} \delta(D_t = d_{t-1} - 1) & d_{t-1} > 0 \\ \mathcal{D}_{s_t} & d_{t-1} = 0 \end{cases} \tag{5}$$

$$E_t \mid s_t \sim \mathcal{O}_{s_t} \tag{6}$$

where $\delta(a, b)$ is the Kronecker product and equals 1 if $a = b$ and 0 otherwise. Given equation 4 and 5, we can write the distribution for $Z_t = \{S_t, D_t\}$

$$\begin{aligned} P(Z_t \mid z_{t-1}) &= P(S_t \mid s_{t-1}, d_{t-1}) P(D_t \mid s_t, d_{t-1}) \\ &= 1 \vee \mathcal{T}_{s_{t-1},.} \mathcal{D}_{s_t} \end{aligned} \tag{7}$$

The joint distribution of an EDHMM given the parameter corresponding to the graphical model can be stated as

$$P(S, D, E \mid \theta) = P(s_0 \mid \mathcal{T}_0) P(d_0 \mid \mathcal{D}_0) \prod_{t=1}^{T} P(s_t \mid s_{t-1}, d_{t-1}, \mathcal{T}) P(d_t \mid s_t, d_{t-1}, \mathcal{D}) P(e_t \mid s_t, \mathcal{O}) \tag{8}$$

Particle Filter Algorithm

input : Observation $e_{1:T}$, importance distribution π, parameter $\theta = \{\mathcal{Z} = \{\mathcal{D}, \mathcal{T}\}, \mathcal{O}\}$, ParticleNumber N
output: Particles $X^{1:N}_{1:T}$, Weights $w^{1:N}_{1:T}$, Weights.Normalized $W^{1:N}_{1:T}$

```
// Initialize particles and weights
```

for $n \leftarrow 1$ **to** N **do**

Sample particle $X_1^n \sim \mathcal{Z}_0$;

Compute $w_1^n(x_1^n) = \dfrac{\mathcal{O}_{x_1^n}(e_1)\mathcal{Z}_0(x_1^n)}{\pi(x_1^n|e_1)}$

end

Resample $(x_1^n, w_1^n)_{n=1:N}$ *with replacement to get equally weighted particles* $(\tilde{x}_1^n, \frac{1}{N})_{n=1:N}$;

```
// Recursively calculate probabilities of interest
```

for $t \leftarrow 2$ **to** T **do**

for $n \leftarrow 1$ **to** N **do**

1. Sample $X_t^n \sim \pi(X_t \mid \tilde{x}^n_{t-1}, e_t)$;

2. Set $X^n_{1:t} = (\tilde{x}^n_{1:t-1}, x_t^n)$;

3. Calculate weights:

$$w_t^n(X^n_{1:t}) \propto w^n_{t-1}(X^n_{1:t-1}) \frac{\mathcal{O}_{x_t^n}(e_t)\mathcal{Z}_{x^n_{t-1}, x^n_t}}{\pi(x_t^n \mid \tilde{x}^n_{t-1}, e_t)}$$

end

i. Normalize all N weights:

$$W_t^n = \frac{w_t^n(X^n_{1:t})}{\sum_i w_t^i(X^i_{1:t})}$$

ii. Resample $(x^n_{1:t}, w_t^n)_{n=1:N}$ *with replacement to get equally weighted particles* $(\tilde{x}^n_{1:t}, \frac{1}{N})_{n=1:N}$;

end

Algorithm 1: General particle filter algorithm

Particle Filter time complexity

Forward backward algorithm in basic HMM: $\mathcal{O}(K^2 T)$, where K is the number of states and T is the time. At each time point t, one needs to evaluate both, the forward and the backward probabilities for all hidden states. If one would not use this iterative procedure and just try a brute force method to find all possible state sequences, one would have a time complexity of $\mathcal{O}(K^T T)$.

Forward backward algorithm in HSMM: In addition to the basic HMM complexity, one needs to truncate the sequence to a minimum and maximum duration, d_{min} and d_{max}. The computational complexity then becomes $\mathcal{O}(K^2(d_{max} - d_{min})^2 T)$, where typically $(d_{max} - d_{min}) >> K$ might go from 0 to T.

Particle Filter algorithm in basic HMM and HSMM: Computational complexity both linear in time T and in number of particles N, so complexity is $\mathcal{O}(NT)$. However, if I would do forward filtering an backward something as well, the complexity would then also be $\mathcal{O}(N^2 T)$. Usually, $N >> K$, but if K is growing (Infinite HMM/HSMM), particle filter might be faster than forward-backward algorithms.

Conditional Particle Filter: is a special case, where a reference trajectory guides the particle filter, making it possible to use very few particles and also use backward smoothing efficiently.

Particle MCMC Algorithm

input : Proposal distribution Q, iterationNumber N,
Particle filter proposal π, particleNumber M,
observation $e_{1:T}$

output: $(\theta^i, Z^i_{1:T})_{i=1:N}$

Initialize θ ;

Run particle filter $\rightarrow$ *get* $\hat{P}(e_{1:T} \mid \theta)$. ;

for $i \leftarrow 1$ **to** N **do**

1. *Propose a new* $\theta^\star$, $\theta^\star \sim Q(\theta^\star \mid \theta)$;
2. *Run particle filter* $\rightarrow$ *get* $\hat{P}(e_{1:T} \mid \theta^\star)$ *and* $Z^\star_{1:T}$. ;
3. *Accept the pair* $(\theta^\star, Z^\star_{1:T})$ *with probability:*

$$min(1, \frac{\hat{P}(e_{1:T} \mid \theta^\star)}{\hat{P}(e_{1:T} \mid \theta)} \frac{P(\theta^\star)}{P(\theta)} \frac{Q(\theta \mid \theta^\star)}{Q(\theta^\star \mid \theta)})$$

4. *If accepted, set* $\hat{P}(e_{1:T} \mid \theta) = \hat{P}(e_{1:T} \mid \theta^\star)$ *and* $\theta = \theta^\star$.

end

Algorithm 2: Particle Metropolis Hastings algorithm

Hamiltonian Monte Carlo Primer

(1) To draw from posterior of interest, introduce auxiliary momentum variable ρ and draw from the joint density $p(\rho, \theta) = p(\rho \mid \theta)p(\theta)$. Usually, ρ does not depend on θ. $p(\rho, \theta)$ define a Hamiltonian

$$\begin{aligned} H(\rho, \theta) &= -logp(\rho, \theta) \\ &= -logp(\rho \mid \theta) - logp(\theta) \\ &= T(\rho \mid \theta) + V(\theta), \end{aligned} \tag{9}$$

where $T(\rho \mid \theta)$ is called "kinetic energy", and $V(\theta)$ "potential energy" ($\propto$ - log posterior).
(2) Joint system $\{\rho, \theta\}$ evolves via Hamiltonian equations

$$\begin{aligned} \frac{d\theta}{dt} &= +\frac{\partial H}{\partial \rho} = +\frac{\partial T}{\partial \rho} \\ \frac{d\rho}{dt} &= -\frac{\partial H}{\partial \theta} = -\frac{\partial T}{\partial \theta} - \frac{\partial V}{\partial \theta} \end{aligned} \tag{10}$$

(3) To solve two-state differential equations in (2), can use, e.g., leapfrog integrator.

- (a) First, sample $\rho \sim MvNormal(0, M)$
- (b) Alternate half-step updates of the momentum and full-step updates of the position L times (for some discretization size ϵ):
  - (i) $\rho \leftarrow \rho - \frac{\epsilon}{2}\frac{\partial V}{\partial \theta}$
  - (ii) $\theta \leftarrow \theta + \epsilon M^{-1}\rho$
  - (iii) $\rho \leftarrow \rho - \frac{\epsilon}{2}\frac{\partial V}{\partial \theta}$
- (c) half-step back for the momentum variable ρ
- (d) Apply a Metropolis acceptance step to account for numerical errors, which can be stated via the Hamiltonians,

$$\alpha = min(1, exp(H(\rho, \theta) - H(\rho^*, \theta^*))) \tag{11}$$

(4) Once (3) is finished, discard momentum variable to have a draw from the posterior via HMC. Many different variations available, also ways to automate tuning of ϵ or L (or both if integration time dynamic) and M.

References

Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society Series B*, 72(3):269–342.

Andrieu, C. and Roberts, G. O. (2009). The pseudo-marginal approach for efficient monte carlo computations. *Ann. Statist.*, 37(2):697–725.

Baum, L. and Petrie, T. (1966). Statistical inference for probabilistic functions of finite state markov chains. *The Annals of Mathematical Statistics*, 37:1554–1563.

References ii

Chopin, N. (2002). A sequential particle filter method for static models. *Biometrika*, 89(3):539–552.

Chopin, N., Jacob, P. E., and Papaspiliopoulos, O. (2013). Smc2: an efficient algorithm for sequential analysis of state-space models.

Doucet, A. and Johansen, A. (2011). A tutorial on particle filtering and smoothing: Fifteen years later.

Jordan, A., Krüger, F., and Lerch, S. (2019). Evaluating probabilistic forecasts with scoringrules. *Journal of Statistical Software, Articles*, 90(12):1–37.

Lindsten, F., Bunch, P., Singh, S. S., and Schön, T. B. (2015). Particle ancestor sampling for near-degenerate or intractable state transition models.

References iii

Lindsten, F., Jordan, M. I., and Schön, T. B. (2014). Particle gibbs with ancestor sampling.

McNeil, A., Frey, R., and Embrechts, P. (2005). *Quantitative Risk Management: Concepts, Techniques and Tools*. Princeton University Press, USA.

Yu, S. (2016). *Hidden Semi-Markov Models: Theory, Algorithms and Applications*. Elsevier, Boston.

Yu, S.-Z. (2010). Hidden semi-markov models. *Artificial Intelligence*, 174(2):215–243. Special Review Issue.