Nonlinear system identification using sequential Monte Carlo methods

Part 4 – Identification strategy 2 (Data augmentation)

Thomas Schön
Division of Systems and Control
Department of Information Technology
Uppsala University.
Email: thomas.schon@it.uu.se, www: user.it.uu.se/~thosc112

Outline – Part 4

Aim: Show how SMC can be used to implement identification strategy 2 – Data augmentation.

1. Summary of Part 3
2. Identification strategy 2 – Data augmentation
3. Maximum likelihood via Expectation Maximization (EM)
4. Sampling state trajectories using Markov kernels
5. Bayesian identification using Gibbs sampling
6. Some of our current research activities (if there is time)

Summary of Part 3

We introduced an auxiliary variable
\[ u = (x_{1:T}, a_{2:T}), \quad u \sim \psi_\theta(u \mid y_{1:T}). \]
and considered an extended target distribution
\[ \phi(\theta, u \mid y_{1:T}) = \frac{\hat{p}_\theta(u_{1:T})\psi_\theta(u \mid y_{1:T})p(\theta)}{p(y_{1:T})}. \]
Could construct a standard MH algorithm that operates in the non-standard extended space \( \Theta \times X^{NT} \times \{1, \ldots, N\}^{N(T-1)} \).
\( p(\theta \mid y_{1:T}) \) is recovered exactly as the marginal of the extended target distribution \( \phi(\theta, u \mid y_{1:T}) \), despite the fact that we employ an SMC approximation of the likelihood using a finite number of particles \( N \).

Identification strategy 2 – Data augmentation

Motivation: If we had access to the complete likelihood
\[ p_\theta(x_{1:T}, y_{1:T}) = \mu_\theta(x_1) \prod_{t=1}^{T} g_\theta(y_t \mid x_t) \prod_{t=1}^{T-1} f_\theta(x_{t+1} \mid x_t) \]
the problem would be much easier.

Key idea: Treat the state sequence \( x_{1:T} \) as an auxiliary variable that is estimated together with the parameters \( \theta \).

The data augmentation strategy breaks the original problem into two new and closely linked problems.

Intuitively: Alternate between updating \( \theta \) and \( x_{1:T} \).
Maximum likelihood via EM

1. **Expectation maximization (EM)** employs the complete likelihood $p_\theta(x_{1:T}, y_{1:T})$ as a substitute for the observed likelihood $p_\theta(y_{1:T})$.

They are of course related,

$$p_\theta(x_{1:T}, y_{1:T}) = p_\theta(x_{1:T} | y_{1:T}) p_\theta(y_{1:T}).$$

It can be shown that by iteratively maximizing

$$Q(\theta, \theta[k]) = \int \log p_\theta(x_{1:T}, y_{1:T}) p_{\theta[k]}(x_{1:T} | y_{1:T}) dx_{1:T}$$

$$= E_{\theta[k]} \left[ \log p_\theta(x_{1:T}, y_{1:T}) | y_{1:T} \right],$$

w.r.t. $\theta$, the obtained sequence $\{\theta[k]\}_{k \geq 1}$ will result in a monotonical increase in likelihood values.

**SMC** is used to approximate the JSD $p_{\theta[k]}(x_{1:T} | y_{1:T})$.

---

Using EM and particle smoothing together

**Algorithm 1** EM for identifying nonlinear dynamical systems

1. **Initialise**: Set $k = 1$ and choose an initial $\theta_1$.

2. **While** not converged do:

   (a) **Expectation (E) step**: Compute

   $$Q(\theta, \theta_k) = \int \log p_\theta(x_{1:T}, y_{1:T}) p_{\theta_k}(x_{1:T} | y_{1:T}) dx_{1:T}$$

   using **sequential Monte Carlo** (particle smoother).

   (b) **Maximization (M) step**: Compute $\theta_{k+1} = \arg \max_{\theta} Q(\theta, \theta_k)$

   (c) $k \leftarrow k + 1$

---

Ex – blind Wiener identification (I/III)

- Second order LGSS model with complex poles.
- Results obtained using $T = 1000$ samples.
- Employ the EM-PS with $N = 100$ particles.
- The plots are based on 100 realisations of data.
- Nonlinearities (dead zone and saturation) shown on next slide.

**Id. problem**: Find $\mathcal{L}$, $\beta$, $r_1$, and $r_2$ based on $\{y_{1,1:T}, y_{2,1:T}\}$.

---

Ex – blind Wiener identification (II/III)

- Bode plot of estimated mean (black), true system (red) and the result for all 100 realisations (gray).
Ex – blind Wiener identification (III/III)

Estimated mean (black), true static nonlinearity (red) and the result for all 100 realisations (gray).


Outline – Part 4

Aim: Show how SMC can be used to implement identification strategy 2 – Data augmentation.

1. Summary of Part 3
2. Identification strategy 2 – Data augmentation
3. Maximum likelihood via Expectation Maximization (EM)
4. Sampling state trajectories using Markov kernels
5. Bayesian identification using Gibbs sampling
6. Some of our current research activities (if there is time)

Gibbs sampler for SSMs

Aim: Compute \( p(\theta, x_{1:T} | y_{1:T}) \).

MCMC: Gibbs sampling ( blocked) for SSMs amounts to iterating

- Draw \( \theta[m] \sim p(\theta | x_{1:T}[m-1], y_{1:T}) \); \( \text{OK!} \)
- Draw \( x_{1:T}[m] \sim p(x_{1:T} | \theta[m], y_{1:T}) \). \( \text{Hard!} \)

Problem: \( p(x_{1:T} | \theta[m], y_{1:T}) \) not available!

Idea: Approximate \( p(x_{1:T} | \theta[m], y_{1:T}) \) using a sequential Monte Carlo method!

Sampling based on SMC

With \( P(x_{1:T}^* = x_{1:T}^i) \propto w_i^t \) we get, \( x_{1:T}^* \sim p(x_{1:T} | \theta, y_{1:T}) \).
Problems and a solution

Problems with this approach,
- Based on a PF ⇒ approximate sample.
- Does not leave $p(x_{1:T} | \theta, y_{1:T})$ invariant!
- Relies on large $N$ to be successful.
- A lot of wasted computations.

To get around these problems,

Use a conditional particle filter (CPF). One pre-specified reference trajectory is retained throughout the sampler.


PMCMC

The idea underlying PMCMC is to make use of a certain SMC sampler to construct a Markov kernel leaving the joint smoothing distribution $p(x_{1:T} | \theta, y_{1:T})$ invariant.

This Markov kernel is then used in a standard MCMC algorithm (e.g. Gibbs, results in the Particle Gibbs (PG)).

SMC is used to build an MCMC kernel with $p(x_{1:t} | \theta, y_{1:t})$ as its stationary distribution without introducing any systematic errors!

Conditional particle filter (CPF)

Let $x'_{1:T} = (x'_1, \ldots, x'_T)$ be a fixed reference trajectory.
- At each time $t$, sample only $N - 1$ particles in the standard way.
- Set the $N^{th}$ particle deterministically: $x^N_t = x'_t$.

CPF causes us to degenerate to the something that is very similar to the reference trajectory, resulting in slow mixing.

Three SMC samplers

Three SMC samplers leaving $p(x_{1:T} | \theta, y_{1:T})$ invariant:

1. Conditional particle filter (CPF)  

2. CPF with backward simulation (CPF-BS)  


3. CPF with ancestor sampling (CPF-AS)  

**CPF vs. CPF-AS – motivation**

BS is problematic for models with more intricate dependencies.

**Reason:** Requires complete trajectories of the latent variable in the backward sweep.

**Solution:** Modify the computation to achieve the same effect as BS, but without an explicit backwards sweep.

**Implication:** Ancestor sampling opens up for inference in a wider class of models, e.g. non-Markovian SSMs, PGMs and BNP models.

Ancestor sampling is conceptually similar to backward simulation, but instead of separate forward and backward sweeps, we achieve the same effect in a single forward sweep.

---

**CPF-AS – algorithm**

Algorithm CPF-AS, conditioned on $x'_{1:T}$

1. Initialize ($t = 1$):
   (a) Draw $x_i^t \sim r_{\theta,1}(x_i^t)$, for $i = 1, \ldots, N - 1$.
   (b) Set $x_i^N = x_i^1$.
   (c) Set $w_i^t = W_{\theta,1}(x_i^1)$.
2. For $t = 2$ to $T$ do:
   (a) Draw $a_i^t \sim C(\{w_i^t\}_{j=1}^N)$, for $i = 1, \ldots, N - 1$.
   (b) Draw $x_i^t \sim r_{\theta,t}(x_i^t \mid x_{i,t-1}^{a_i^{t-1}})$, for $i = 1, \ldots, N - 1$.
   (c) Set $x_i^N = x_i^t$.
   (d) Draw $a_i^N$ with $\mathbb{P}(a_i^N = i) \propto w_i^{t-1}_{j=1} \gamma_{\theta,t-1}(x_{i,t-1}^{a_i^{t-1}}, x_{i,t}^a)$.  
   (e) Set $x_i^t = \{x_{i,t-1}^{a_i^{t-1}}, x_i^t\}$ and $w_i^t = W_{\theta,t}(x_i^t)$.

---

**CPF-AS – intuition**

Let $x'_{1:T} = (x'_1, \ldots, x'_T)$ be a fixed reference trajectory.
- At each time $t$, sample only $N - 1$ particles in the standard way.
- Set the $N^\text{th}$ particle deterministically: $x_i^N = x'_i$.
- Generate an artificial history for $x_i^N$ by ancestor sampling.

CPF-AS causes us to degenerate to something that is very different from the reference trajectory, resulting in better mixing.

---

**The PGAS Markov kernel**

1. Run CPF-AS($x'_{1:T}$) targeting $p(x_{1:T} \mid \theta, y_{1:T})$.
2. Sample $x_{1:T}^*$ with $\mathbb{P}(x_{1:T}^* = x'_{1:T}) \propto w_T$.

- Maps $x_{1:T}^*$ stochastically into $x_{1:T}^*$.
- Implicitly defines an ergodic Markov kernel ($P_\theta^N$) referred to as the PGAS (particle Gibbs with ancestor sampling) kernel.

**Theorem**

For any number of particles $N \geq 1$ and $\theta \in \Theta$, the PGAS kernel $P_\theta^N$ leaves $p(x_{1:T} \mid \theta, y_{1:T})$ invariant,

$$p(dx_{1:T}^* \mid \theta, y_{1:T}) = \int P_\theta^N(x_{1:T}^*, dx_{1:T})p(dx_{1:T}^* \mid \theta, y_{1:T})$$
**Toy example – stochastic volatility (I/II)**

Consider the stochastic volatility model,
\[ x_{t+1} = 0.9x_t + w_t, \quad w_t \sim \mathcal{N}(0, \theta), \]
\[ y_t = e_t \exp \left( \frac{1}{2} x_t \right), \quad e_t \sim \mathcal{N}(0, 1). \]

Let us study the ACF for the estimation error, \( \hat{\theta} - \mathbb{E} [\theta | y_{1:T}] \)

**Example – semiparametric Wiener model**

Parametric LGSS and a nonparametric static nonlinearity:
\[ x_{t+1} = \begin{pmatrix} A & B \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_t \\ u_t \end{pmatrix} + v_t, \quad v_t \sim \mathcal{N}(0, Q), \]
\[ z_t = Cx_t, \]
\[ y_t = g(z_t) + e_t, \quad e_t \sim \mathcal{N}(0, R). \]
Example – semiparametric Wiener model

“Parameters”: $\theta = \{A, B, Q, g(\cdot), r\}$.

Bayesian model specified by priors
- Conjugate priors for $\Gamma = [A B], Q$ and $r$,
  - $p(\Gamma, Q) =$ Matrix-normal inverse-Wishart
  - $p(r) =$ inverse-Wishart
- Gaussian process prior on $g(\cdot)$,
  $g(\cdot) \sim \mathcal{GP}(z, k(z, z'))$.

Inference using PGAS with $N = 15$ particles. We ran 15 000 MCMC iterations and discarded 5 000 as burn-in.

Some of our current research activities

Joint work with (alphabetical order):

John Aston (University of Cambridge), Alexandre Bouchard-Côté (University of British Columbia), Pete Bunch (University of Cambridge), Adam M. Johansen (University of Warwick), Bonnie Kirkpatrick (University of Miami), Fredrik Lindsten (University of Cambridge), Christian A. Naesseth (Linköping University), Sumeetpal S. Singh (University of Cambridge), Arno Solin (Aalto University), Andreas Svensson (Uppsala University), Simo Särkkä (Aalto University) and

Inference in probabilistic graphical models

Constructing an artificial sequence of intermediate target distributions for an SMC sampler is a powerful (and quite possibly underutilized) idea.
SMC in high dimensions

The bootstrap PF suffers from weight collapse in high-dimensional settings.

This degeneracy can be reduced by using so-called fully adapted proposals.

We can mimic the efficient fully adapted proposals for arbitrary latent spaces and structures in high-dimensional models.

Approximations the proposal distribution and use a nested coupling of multiple SMC samplers and backward simulators.


A nonparametric model – GP-SSM

Consider the Gaussian Process SSM (GP-SSM):

\[ x_{t+1} = f(x_t) + w_t, \quad \text{s.t.} \quad f(x) \sim \mathcal{GP}(0, \kappa, f(x, x')) \]
\[ y_t = g(x_t) + e_t, \quad \text{s.t.} \quad g(x) \sim \mathcal{GP}(0, \kappa, g(x, x')) \]

The model functions \( f \) and \( g \) are assumed to be realizations from a Gaussian process prior. The noise processes \( w_t \) and \( e_t \) are zero-mean Gaussian with covariace matrices \( Q \) and \( R \), respectively.

We will identify this model by inferring the posterior distribution

\[ p(f, g, Q, R, \theta | y_{1:T}) \]

The PGAS construction is key.


Near-degenerate/intractable state transition

Two model classes where state-of-the-art algorithms still struggle:

1. models where the state transition kernel is (nearly) degenerate, i.e. (nearly) concentrated on a low-dimensional manifold.
2. models where point-wise evaluation of the state transition density is intractable.

The main difficulty is that the backward kernel will also be (nearly) degenerate or intractable, effectively ruling out the standard forward-backward-based methods.

We introduce a key construction allowing us to replace simulation from the (problematic) backward kernel, with a simulation from a joint density over a subset of future state variables.


The aim of this course

The aim of this course is to provide an introduction to the theory and application of sequential Monte Carlo (SMC) methods for nonlinear system identification (and state estimation) problems.

When it comes to SMC we will focus on the particle filter.

After this course you should be able to derive your own SMC based algorithms allowing you to solve nonlinear system identification and state estimation problems.

If you need inspiration I have provided some exercises here

user.it.uu.se/~thosc112/courses.html
Conclusions

1. Provided two key strategies and four different approaches for system identification in nonlinear SSMs.
2. Showed how SMC can be used to implement these strategies.
3. (Hopefully) conveyed the intuition underlying SMC.
4. SMC is applicable to many problems, not just SSMs via PF.

Manuscript is also available (just ask me for a draft)


PhD course on the topic is available here

user.it.uu.se/~thosc112/CIDS.html

There will be an invited session on SMC at SYSID in Beijing (China) in October.

**Fast moving research area offering lots of opportunities!**