Learning dynamical systems using particle filters

Thomas Schöen
Division of Systems and Control
Department of Information Technology
Uppsala University

UPPSALA UNIVERSITET

Joint work with (alphabetical order): Roger Frigola (University of Cambridge), Fredrik Gustafsson (Linköping University), Michael I. Jordan (UC Berkeley), Johan Kihlberg (Semcon), Fredrik Lindsten (Linköping University), Lennart Ljung (Linköping University), Brett Ninness (University of Newcastle, Australia), Per Johan Nordlund (Saab), Carl E. Rasmussen (University of Cambridge), Simon Tegelid (Xdin), Adrian Wills (MRA, Newcastle, Australia).

Thomas Schöen (user.it.uu.se/thosc112), Learning dynamical systems using particle filters
Seminart at the Division of Scientific Computing, Uppsala University, November 20, 2013.
Dynamical systems are everywhere!

Some of the dynamical systems we have been working with,

We first have to learn the models. Then we can use them.
The sequential Monte Carlo samplers are fundamental to both the maximum likelihood and the Bayesian approaches.
1. Probabilistic models of dynamical systems
2. State inference
3. Sequential Monte Carlo (SMC), the particle filter
   a) Key idea
   b) indoor localization example
   c) UAV localization example
4. Learning dynamical models
   a) Maximum Likelihood (ML) identification (very brief)
   b) Bayesian identification ((P)MCMC)
   c) Particle Gibbs with ancestor sampling (PG-AS)

The sequential Monte Carlo samplers are fundamental to both the maximum likelihood and the Bayesian approaches.
Basic representation: Two discrete-time stochastic processes,

- \( \{x_t\}_{t \geq 1} \) representing the state of the system.
- \( \{y_t\}_{t \geq 1} \) representing the measurements from the sensors.

The probabilistic model is described using two \( (f \text{ and } g) \) probability density functions (PDFs):

\[
\begin{align*}
  x_{t+1} | x_t & \sim f_\theta(x_{t+1} | x_t, u_t), \\
  y_t | x_t & \sim g_\theta(y_t | x_t).
\end{align*}
\]

Model = PDF

This type of model is referred to as a **state space model (SSM)** or a **hidden Markov model (HMM)**.
Aim: Compute a probabilistic representation of our knowledge of the state, based on information that is present in the measurements.

The filtering PDF

\[ p(x_t \mid y_{1:t}) \]

provides a representation of the uncertainty about the state at time \( t \), given all the measurements up to time \( t \).

The obvious question is now, how do we compute this object?

\[
\begin{align*}
\nonumber p(x_t \mid y_{1:t}) &= p(x_t \mid y_t, y_{1:t-1}) \\
&= \frac{p(y_t \mid x_t, y_{1:t-1})p(x_t \mid y_{1:t-1})}{p(y_t \mid y_{1:t-1})}
\end{align*}
\]

\[
= \frac{g(y_t \mid x_t)p(x_t \mid y_{1:t-1})}{p(y_t \mid y_{1:t-1})}.
\]
Apparently we need an expression also for the prediction PDF

\[ p(x_t \mid y_{1:t-1}). \]

Using marginalization we have

\[
p(x_t \mid y_{1:t-1}) = \int p(x_t, x_{t-1} \mid y_{1:t-1}) \, dx_{t-1}
\]

\[
= \int p(x_t \mid x_{t-1}, y_{1:t-1}) p(x_{t-1} \mid y_{1:t-1}) \, dx_{t-1}.
\]

Hence, the prediction PDF is given by

\[
p(x_t \mid y_{1:t-1}) = \int f(x_t \mid x_{t-1}) p(x_{t-1} \mid y_{1:t-1}) \, dx_{t-1}.
\]
We have now showed that for the nonlinear SSM
\[ x_{t+1} \mid x_t \sim f(x_{t+1} \mid x_t), \]
\[ y_t \mid x_t \sim g_\theta(y_t \mid x_t). \]
the uncertain information that we have about the state is captured by
the filtering PDF, which we compute sequentially using a
measurement update
\[
p(x_t \mid y_{1:t}) = \frac{\underbrace{g(y_t \mid x_t)} \cdot \underbrace{p(x_t \mid y_{1:t-1})}}{\underbrace{p(y_t \mid y_{1:t-1})}}
\]
and a time update
\[
p(x_t \mid y_{1:t-1}) = \int \frac{\underbrace{f(x_t \mid x_{t-1})} \cdot \underbrace{p(x_{t-1} \mid y_{1:t-1})}}{\underbrace{p(x_{t-1} \mid y_{1:t-1})}} \, dx_{t-1}.
\]
Consider the following special case (Linear Gaussian State Space (LGSS) model)

\[ x_{t+1} = Ax_t + Bu_t + v_t, \quad v_t \sim \mathcal{N}(0, Q_t), \]
\[ y_t =Cx_t + Du_t + e_t, \quad e_t \sim \mathcal{N}(0, R_t). \]

or, equivalently,

\[ x_{t+1} \mid x_t \sim f(x_{t+1} \mid x_t) = \mathcal{N}(x_{t+1} \mid Ax_t + Bu_t, Q_t), \]
\[ y_t \mid x_t \sim g(y_t \mid x_t) = \mathcal{N}(y_t \mid Cx_t + Du_t, R_t). \]

It is now straightforward to show that the solution to the time update and measurement update equations is given by the Kalman filter,

\[ p(x_t \mid y_{1:t}) = \mathcal{N}(x_t \mid \hat{x}_{t|t}, P_{t|t}), \]
\[ p(x_{t+1} \mid y_{1:t}) = \mathcal{N}(x_{t+1} \mid \hat{x}_{t+1|t}, P_{t+1|t}). \]
**Obvious question:** what do we do in an interesting case, for example when we have a nonlinear model with non-Gaussian noise?

1. Need a general representation of the filtering PDF
2. Try to solve the equations

\[
p(x_t \mid y_{1:t}) = \frac{g(y_t \mid x_t)p(x_t \mid y_{1:t-1})}{p(y_t \mid y_{1:t-1})},
\]

\[
p(x_t \mid y_{1:t-1}) = \int f(x_t \mid x_{t-1})p(x_{t-1} \mid y_{1:t-1})\,dx_{t-1},
\]

as accurately as possible.
1. Probabilistic models of dynamical systems

2. State inference

3. **Sequential Monte Carlo (SMC), the particle filter**
   a) Key idea
   b) indoor localization example
   c) UAV localization example

4. Learning dynamical models
   a) Maximum Likelihood (ML) identification (very brief)
   b) Bayesian identification ((P)MCMC)
   c) Particle Gibbs with ancestor sampling (PG-AS)

The sequential Monte Carlo samplers are fundamental to **both** the maximum likelihood and the Bayesian approaches.
The particle filter provides an approximation of the filtering PDF $p(x_t \mid y_{1:t})$, when the state evolves according to an SSM,

$$
\begin{align*}
x_{t+1} \mid x_t & \sim f_t(x_{t+1} \mid x_t), \\
y_t \mid x_t & \sim g_t(y_t \mid x_t), \\
x_1 & \sim \mu(x_1).
\end{align*}
$$

The particle filter maintains an empirical distribution made up of $N$ samples (particles) $\{x^i_t\}_{i=1}^N$ and corresponding weights $\{w^i_t\}_{i=1}^N$

$$
\hat{p}^N(x_t \mid y_{1:t}) = \sum_{i=1}^N w^i_t \delta_{x^i_t}(x_t).
$$

"Think of each particle as one simulation of the system state. Only keep the good ones."
The particle filter has been around for roughly 20 years.

The use of particle methods for nonlinear system identification started to take off some 5 years ago.

Now this is a very active problem (and solution) within many fields.
Consider a toy 1D localization problem.

**Dynamic model:**

\[ x_{t+1} = x_t + u_t + v_t, \]

where \( x_t \) denotes position, \( u_t \) denotes velocity (known), \( v_t \sim \mathcal{N}(0, 5) \) denotes an unknown disturbance.

**Measurements:**

\[ y_t = h(x_t) + e_t. \]

where \( h(\cdot) \) denotes the world model (here the terrain height) and \( e_t \sim \mathcal{N}(0, 1) \) denotes an unknown disturbance.

The same idea has been used for the Swedish fighter JAS 39 Gripen. Details are available in,


Thomas Schön (user.it.uu.se/thosc112), *Learning dynamical systems using particle filters*

Seminart at the Division of Scientific Computing, Uppsala University, November 20, 2013.
The particle filter – toy problem

Highlights two key capabilities of the PF:

1. Automatically handles an unknown and dynamically changing number of hypotheses.

2. Work with nonlinear/non-Gaussian models.
The particle filter

1. **Resampling**: \( \{x_{i-1}^i, w_{i-1}^i\}_{i=1}^N \rightarrow \{\tilde{x}_{i-1}^i, 1/N\}_{i=1}^N \).

2. **Propagation**: \( x_t^i \sim q_t(x_t \mid \tilde{x}_{t-1}^i) \).

3. **Weighting**: \( w_t^i = W_t(x_t^i, y_t) \).

The result is a new weighted set of particles \( \{x_t^i, w_t^i\}_{i=1}^N \) targeting \( p(x_t \mid y_{1:t}) \).

A systematic way of obtaining approximations that converge


Thomas Schön (user.it.uu.se/thosc112), **Learning dynamical systems using particle filters**

Seminart at the Division of Scientific Computing, Uppsala University, November 20, 2013.
**Aim:** Compute the position of a person moving around indoors using sensors (inertial, magnetometer and radio) located in an ID badge and a map.

Show movie
Aim: Compute the position and orientation of a helicopter by exploiting the information present in Google maps images of the operational area.

Sensor fusion in dynamical systems

Thomas Schön, users.isy.liu.se/rt/schon
The University of British Columbia
Vancouver, Canada

World model
Inference
Dynamic model
Sensor model
Sensors
Camera
Inertial
Barometer
Pose

Learning dynamical systems using particle filters
Seminart at the Division of Scientific Computing, Uppsala University, November 20, 2013.
Example 2 – UAV localization (II/III)

Map over the operational environment obtained from Google Earth.

Manually classified map with grass, asphalt and houses as pre-specified classes.

Image from on-board camera

Extracted superpixels

Superpixels classified as grass, asphalt or house

Three circular regions used for computing class histograms

Thomas Schön (user.it.uu.se/thosc112), Learning dynamical systems using particle filters
Seminart at the Division of Scientific Computing, Uppsala University, November 20, 2013.
Example 2 – UAV localization (III/III)

Show movie


Thomas Schön (user.it.uu.se/thosc112), Learning dynamical systems using particle filters
Seminart at the Division of Scientific Computing, Uppsala University, November 20, 2013.
1. Probabilistic models of dynamical systems
2. State inference
3. Sequential Monte Carlo (SMC), the particle filter
   a) Key idea
   b) indoor localization example
   c) UAV localization example
4. Learning dynamical models
   a) Maximum Likelihood (ML) identification (very brief)
   b) Bayesian identification ((P)MCMC)
   c) Particle Gibbs with ancestor sampling (PG-AS)

The sequential Monte Carlo samplers are fundamental to both the maximum likelihood and the Bayesian approaches.
A state space model (SSM) consists of a Markov process \( \{x_t\}_{t \geq 1} \) and a measurement process \( \{y_t\}_{t \geq 1} \), related according to

\[
\begin{align*}
x_{t+1} \mid x_t & \sim f_t(x_{t+1} \mid x_t), \\
y_t \mid x_t & \sim g_t(y_t \mid x_t), \\
x_1 & \sim \mu(x_1).
\end{align*}
\]

We observe

\[
y_{1:T} \triangleq \{y_1, \ldots, y_T\},
\]

(leaving the latent variables \( x_{1:T} \) unobserved).

**Identification problem:** Find \( f, g, \mu \) (or \( \theta \)) based on \( y_{1:T} \).
Alternate between updating $\theta$ and updating $x_{1:T}$.

Frequentists:
- Find $\hat{\theta}_{\text{ML}} = \arg\max_{\theta} p_{\theta}(y_{1:T})$.
- Use e.g. the expectation maximization (EM) algorithm.

Bayesians:
- Find $p(\theta \mid y_{1:T})$.
- Use e.g. Gibbs or Metropolis-Hastings sampling.
Maximum likelihood (ML) amounts to solving,

$$\hat{\theta}^{ML} = \arg \max_{\theta} \log p_{\theta}(y_{1:T}) = \arg \max_{\theta} \sum_{t=1}^{T} \log p_{\theta}(y_t \mid y_{1:t-1}),$$

where

$$x_{t+1} \mid x_t \sim f_{\theta,t}(x_{t+1} \mid x_t),$$

$$y_t \mid x_t \sim g_{\theta,t}(y_t \mid x_t),$$

$$x_1 \sim \mu_{\theta}(x_1).$$

Can be solved by combining the Expectation Maximization (EM) algorithm with a particle smoother.


Thomas Schön (user.it.uu.se/thosc112), *Learning dynamical systems using particle filters*  
Seminart at the Division of Scientific Computing, Uppsala University, November 20, 2013.
Consider a Bayesian SSM ($\theta$ is now a random variable with a prior density $p(\theta)$)

$$
\begin{align*}
  x_{t+1} | x_t & \sim f_{\theta,t}(x_{t+1} | x_t), \\
y_t | x_t & \sim g_{\theta,t}(y_t | x_t), \\
x_1 & \sim \mu_\theta(x_1), \\
\theta & \sim p(\theta).
\end{align*}
$$

**Identification problem**: Compute the posterior $p(\theta, x_{1:T} | y_{1:T})$, or one of its marginals.

The **key challenge** is that there is no closed form expression available for the posterior.

Markov chain Monte Carlo (MCMC) methods allow us to generate samples from a target distribution by simulating a Markov chain.

**Gibbs sampling** (blocked) for SSMs amounts to iterating

- Draw $\theta[m] \sim p(\theta | x_{1:T}[m-1], y_{1:T})$,
- Draw $x_{1:T}[m] \sim p(x_{1:T} | \theta[m], y_{1:T})$.

The above procedure results in a Markov chain,

$$\{\theta[m], x_{1:T}[m]\}_{m \geq 1}$$

with $p(\theta, x_{1:T} | y_T)$ as its stationary distribution!
What would a Gibbs sampler for a general nonlinear/non-Gaussian SSM look like?

- Draw $\theta[m] \sim p(\theta | x_{1:T}[m-1], y_{1:T})$; \hspace{1cm} \text{OK!}
- Draw $x_{1:T}[m] \sim p(x_{1:T} | \theta[m], y_{1:T})$. \hspace{1cm} \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 the PF (I/II)
Sampling based on the PF (II/II)

With \( P(x'_{1:T} = x^i_{1:T}) \propto w^i_T \) we get, \( x'_{1:T} \approx p(x_{1:T} | \theta, y_{1:T}) \).

Thomas Schöen (user.it.uu.se/thosc112), Learning dynamical systems using particle filters
Seminart at the Division of Scientific Computing, Uppsala University, November 20, 2013.
Problems

Problems with this approach,

- Based on a PF $\Rightarrow$ approximate sample.
- Does not leave $p(\theta, x_{1:T} \mid y_{1:T})$ 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 path is retained throughout the sampler.

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, resulting in the Particle Gibbs (PG)).

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

Thomas Schön (user.it.uu.se/thosc112), *Learning dynamical systems using particle filters*

Seminart at the Division of Scientific Computing, Uppsala University, November 20, 2013.
Theorem

For any \( N \geq 2 \), the procedure;

(i) Run CPF-AS(\( x_{1:T}^* \));

(ii) Sample \( P(x'_{1:T} = x_{1:T}^i) \propto w_T^i \);

defines a Markov kernel on \( X^T \) which leaves \( p(x_{1:T} | \theta, y_{1:T}) \) invariant.

Three additional reasons for using CPF-AS:

1. Significantly improves the mixing compared to CPF.
2. The computational complexity is linear in \( N \).
3. Opens up for non-Markovian models.
Bayesian identification: Gibbs + CPF-AS = PG-AS

Algorithm PG-AS: Particle Gibbs with ancestor sampling

1. Initialize: Set \( \{\theta[0], x_{1:T}[0]\} \) arbitrarily.
2. For \( m \geq 1 \), iterate:
   (a) Draw \( \theta[m] \sim p(\theta \mid x_{1:T}[m - 1], y_{1:T}) \).
   (b) Run CPF-AS\( (x_{1:T}[m - 1]) \), targeting \( p(x_{1:T} \mid \theta[m], y_{1:T}) \).
   (c) Sample with \( P(x_{1:T}[m] = x_{1:T}^i) \propto w_T^i \).

For any number of particles \( N \geq 2 \), the Markov chain \( \{\theta[m], x_{1:T}[m]\}_{m \geq 1} \) has stationary distribution \( p(\theta, x_{1:T} \mid y_{1:T}) \).
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, \( \hat{\theta} - E[\theta | y_{1:T}] \).
Consider the stochastic volatility model,

\[ x_{t+1} = 0.9x_t + w_t, \]
\[ y_t = e_t \exp\left(\frac{1}{2}x_t\right), \]

\[ w_t \sim \mathcal{N}(0, \theta), \]
\[ e_t \sim \mathcal{N}(0, 1). \]

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

- We want the ACF to decay to zero as rapidly as possible (indicates good mixing in the PG sampler).
- Note the superior mixing of PG-AS compared to PG-CPF (already for just $N = 5$ particles!).

Thomas Schön (user.it.uu.se/thosc112), *Learning dynamical systems using particle filters*
Seminart at the Division of Scientific Computing, Uppsala University, November 20, 2013.
Example – semiparametric Wiener model (I/III)

Parametric LGSS and a nonparametric static nonlinearity:

\[
x_{t+1} = \begin{pmatrix} A & B \end{pmatrix} \begin{pmatrix} x_t \\ u_t \end{pmatrix} + \nu_t,
\]

\[
\nu_t \sim \mathcal{N}(0, Q),
\]

\[
z_t = Cx_t.
\]

\[
y_t = g(z_t) + e_t,
\]

\[
e_t \sim \mathcal{N}(0, R).
\]
Everything is learned from the data, by introducing the possibility to switch specific model components on and off.

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

**Bayesian model** specified by priors
- Sparseness prior (ARD) on $\Gamma = [A \ B]$,
- Inverse-Wishart prior on $Q$ and $r$
- Gaussian process prior on $g(\cdot)$,

$$g(\cdot) \sim \mathcal{GP}(z, k(z, z')).$$ 

**Inference** using PG-AS with $N = 15$ particles. $T = 1000$ measurements. We ran 15000 MCMC iterations and discarded 5000 as burn-in.
Example – semiparametric Wiener model (III/III)

Bode diagram of the 4th-order linear system. Estimated mean (dashed black), true (solid black) and 99% credibility intervals (blue).

Static nonlinearity (non-monotonic), estimated mean (dashed black), true (black) and the 99% credibility intervals (blue).

Conclusions

- Probabilistic models of dynamical systems.
- Sequential Monte Carlo introduced via the particle filter.
- EM-PS for ML learning in nonlinear SSMs.
- PG-AS for Bayesian learning in nonlinear SSMs.
- The conditional particle filter defines a kernel on $X^T$ leaving $p_\theta(x_{1:T} \mid y_{1:T})$ invariant.

There is a lot of interesting research that remains to be done!!

- We are working on a book project,
  Send me an e-mail if you are interested in a draft.
- PhD course: [user.it.uu.se/~thosc112/CIDS.html](http://user.it.uu.se/~thosc112/CIDS.html)
Some references

Forthcoming book (includes all material used in this seminar)

Novel introduction of PMCMC (very nice paper!)

Self-contained introduction to BS and AS (not limited to SSMs)

PG-AS (and the Wiener identification example)

ML identification of nonlinear SSMs (and Wiener example)

Bayesian inference using Gaussian processes

MATLAB code is available from our web-site. Thomas Schön (user.it.uu.se/ thosc112), *Learning dynamical systems using particle filters* Seminart at the Division of Scientific Computing, Uppsala University, November 20, 2013.