Ancestor sampling in state space models and beyond

"Ancestor sampling is a way of exploiting backward simulation ideas without needing an explicit backward pass."

UPPSALA
UNIVERSITET

Thomas Schön

Division of Systems and Control
Department of Information Technology
Uppsala University, Sweden

Joint work with (alphabetical order): **Michael I. Jordan** (UC Berkeley), **Fredrik Lindsten** (University of Cambridge) and **Christian A. Naesseth** (Linköping University).

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.

Learning nonlinear dynamical systems from data

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

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

$$\begin{aligned} x_{t+1} \mid x_t &\sim f_{\theta,t}(x_{t+1} \mid x_t), \\ y_t \mid x_t &\sim g_{\theta,t}(y_t \mid x_t), \\ x_1 &\sim \mu_\theta(x_1). \end{aligned}$$

We observe

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

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

Learning problem: Find f, g, μ (or θ) based on $y_{1:T}$.

Strategies for inference in latent variable models

Alternate between updating θ and updating $x_{1:T}$.

Frequentists:

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

Bayesians:

- Find $p(\theta \mid y_{1:T})$.
- Use e.g. Gibbs sampling.

Frequentists + Bayesians:

- Use e.g. particle Gibbs together with stochastic approx. EM.

Outline

1. Maximum Likelihood (ML) learning
    a) Problem formulation
    b) EM and a particle smoother
2. Bayesian learning
    a) Problem formulation
    b) Gibbs sampling
3. Sequential Monte Carlo (SMC)
4. **Particle Gibbs with ancestor sampling (PGAS)**
5. **SMC and PGAS for graphical models**

The sequential Monte Carlo samplers are fundamental to **both** the ML and the Bayesian approaches.

Problem formulation – ML

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

$$\begin{aligned} x_{t+1} \mid x_t &\sim f_{\theta,t}(x_{t+1} \mid x_t), \\ y_t \mid x_t &\sim g_{\theta,t}(y_t \mid x_t), \\ x_1 &\sim \mu_\theta(x_1). \end{aligned}$$

Learning problem: Find θ based on $y_{1:T}$.

ML amounts to solving,

$$\widehat{\theta}^{\text{ML}} = \arg\max_{\theta} \log p_\theta(y_{1:T})$$

where the log-likelihood function is given by

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

EM – the algorithm

```
Algorithm 1 EM – learning nonlinear dynamical systems
1. Initialise: Set $i = 1$ and choose an initial $\theta^1$.
2. While not converged do:
    (a) Expectation (E) step: Compute
        $$\begin{aligned} \mathcal{Q}(\theta, \theta^i) &= \mathrm{E}_{\theta^i}\left[\log p_\theta(x_{1:T}, y_{1:T}) \mid y_{1:T}\right] \\ &= \int \log p_\theta(x_{1:T}, y_{1:T}) \underbrace{p_{\theta^i}(x_{1:T} \mid y_{1:T})} \mathrm{d}x_{1:T} \end{aligned}$$
        using PS (backward simulation or ancestor sampling).
    (b) Maximization (M) step: Compute $\theta^{i+1} = \arg\max_{\theta \in \Theta} \mathcal{Q}(\theta, \theta^i)$
    (c) $i \leftarrow i + 1$
```

Thomas B. Schön, Adrian Wills and Brett Ninness. **System Identification of Nonlinear State-Space Models**. *Automatica*, 47(1):39-49, January 2011.

Problem formulation – Bayesian

Consider a Bayesian SSM (θ is now a r.v. with a prior density $p(\theta)$)

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

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

Key challenge: There is no closed form expression available.

Gibbs sampler for SSMs (I/II)

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

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

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

The above procedure results in a Markov chain,

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

with $p(\theta, x_{1:T} \mid y_T)$ as its stationary distribution!

Gibbs sampler for SSMs (II/II)

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

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

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

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

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

Outline

1. Maximum Likelihood (ML) learning
    a) Problem formulation
    b) Solution using EM and a particle smoother
2. Bayesian learning
    a) Problem formulation
    b) Gibbs sampling
3. **Sequential Monte Carlo (SMC)**
4. Particle Gibbs with ancestor sampling (PGAS)
5. SMC and PGAS for graphical models

The sequential Monte Carlo samplers are fundamental to **both** the ML and the Bayesian approaches.

Sequential Monte Carlo (SMC)

Approximate a **sequence** of probability distributions on a sequence of probability spaces of **increasing dimension**.

Let $\{\gamma_{\theta,t}(x_{1:t})\}_{t\geq 1}$ be a sequence of unnormalized densities and

$$\bar{\gamma}_{\theta,t}(x_{1:t}) = \frac{\gamma_{\theta,t}(x_{1:t})}{Z_{\theta,t}}$$

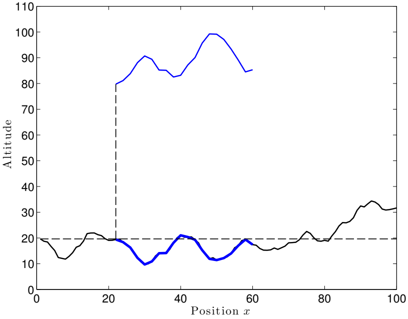
Ex. (SSM)

$$\bar{\gamma}_{\theta,t}(x_{1:t}) = p_\theta(x_{1:t} \mid y_{1:t}), \qquad \gamma_{\theta,t}(x_{1:t}) = p_\theta(x_{1:t}, y_{1:t}),$$

$$Z_{\theta,t} = p_\theta(y_{1:t}).$$

SMC – solving a toy problem

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:

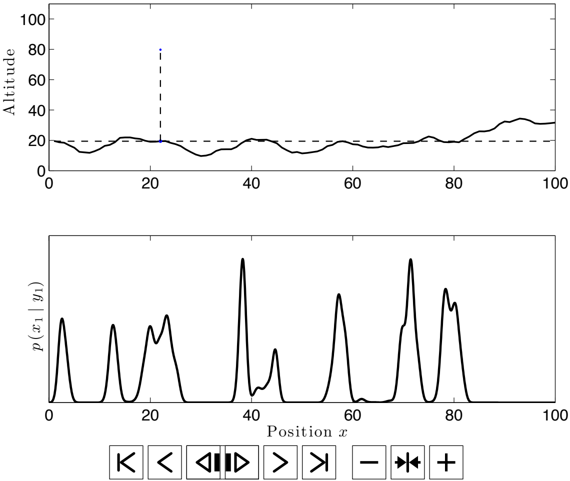
$$y_t = h(x_t) + e_t.$$

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, Fredrik Gustafsson, and Per-Johan Nordlund. **Marginalized particle filters for mixed linear/nonlinear state-space models**. *IEEE Transactions on Signal Processing*, 53(7):2279-2289, July 2005.

SMC – solving a toy problem

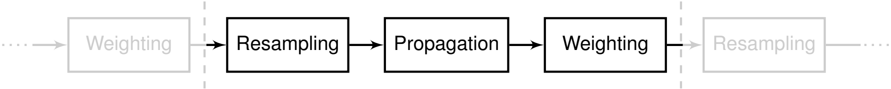


Highlights two **key capabilities** of SMC:

1. Automatically handles an unknown and dynamically changing number of hypotheses.
2. Work with nonlinear/non-Gaussian models.

$$p(x_t \mid y_{1:t}) \approx \sum_{i=1}^{N} w_t^i \delta_{x_t^i}(x_t)$$

Sequential Monte Carlo (SMC)

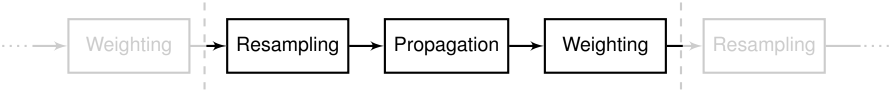


SMC = resampling + sequential importance sampling

1. **Resampling:** $\mathbb{P}(a_t^i = j) = w_{t-1}^j / \sum_l w_{t-1}^l$.
2. **Propagation:** $x_t^i \sim r_{\theta,t}(x_t \mid x_{1:t-1}^{a_t^i})$ and $x_{1:t}^i = \{x_{1:t-1}^{a_t^i}, x_t^i\}$.
3. **Weighting:** $w_t^i = W_{\theta,t}(x_{1:t}^i) = \frac{\gamma_{\theta,t}(x_{1:t}^i)}{\gamma_{\theta,t-1}(x_{1:t-1}^i) r_{\theta,t}(x_t^i | x_{1:t-1}^i)}$.

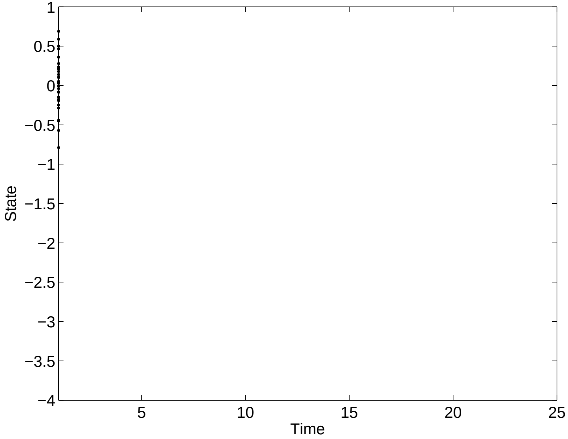
The result is a new weighted set of particles $\{x_{1:t}^i, w_t^i\}_{i=1}^N$.

Sequential Monte Carlo (SMC)



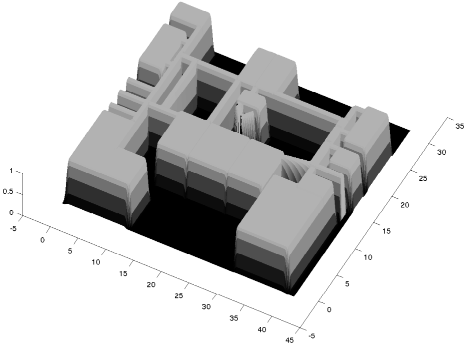
```
Algorithm SMC (each step is for $i = 1, \dots N$)
1. Initialize ($t = 1$):
    (a) Draw $x_1^i \sim r_{\theta,1}(x_1^i)$.
    (b) Set $w_1^i = W_{\theta,1}(x_1^i)$.
2. For $t = 2$ to $T$ do:
    (a) Draw $a_t^i \sim \text{Cat}(\{w_t^j\}_{j=1}^N)$.
    (b) Draw $x_t^i \sim r_{\theta,t}(x_t \mid x_{1:t-1}^{a_t^i})$ and set $x_{1:t}^i = \{x_{1:t-1}^{a_t^i}, x_t^i\}$.
    (c) Set $w_t^i = W_{\theta,t}(x_{1:t}^i)$.
```

SMC suffers from path degeneracy (illustration)



SMC application example – indoor localization

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

Outline

1. Maximum Likelihood (ML) learning
   a) Problem formulation
   b) Solution using EM and a particle smoother
2. Bayesian learning
   a) Problem formulation
   b) Gibbs sampling
3. Sequential Monte Carlo (SMC)
4. **Particle Gibbs with ancestor sampling (PGAS)**
5. SMC and PGAS for graphical models

The sequential Monte Carlo samplers are fundamental to **both** the ML and the Bayesian approaches.

Recall – Gibbs sampler for SSMs

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

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

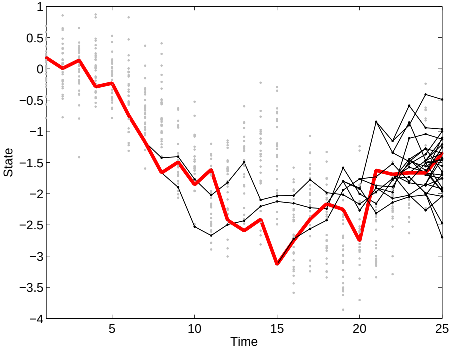
- Draw $\theta[m] \sim p(\theta \mid x_{1:T}[m-1], y_{1:T})$; **OK!**
- Draw $x_{1:T}[m] \sim p(x_{1:T} \mid \theta[m], y_{1:T})$. **Hard!**

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

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

Sampling based on SMC

With $\mathbb{P}(x^{\star}_{1:T} = x^{i}_{1:T}) \propto w^{i}_{T}$ we get, $x^{\star}_{1:T} \overset{\text{approx.}}{\sim} p(x_{1:T} \mid \theta, y_{1:T})$.



Problems and a solution

Problems with this approach,

- Based on a PF ⇒ approximate sample.
- Does not leave $p(x_{1:T} \mid \theta, y_{1:T})$ 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.

Christophe Andrieu, Arnaud Doucet and Roman Holenstein, **Particle Markov chain Monte Carlo methods**, *Journal of the Royal Statistical Society: Series B*, 72:269-342, 2010.

Particle Markov chain Monte Carlo (PMCMC)

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

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

For a self-contained introduction (focused on BS and AS),

Fredrik Lindsten and Thomas B. Schön, **Backward simulation methods for Monte Carlo statistical inference**, *Foundations and Trends in Machine Learning*, 6(1):1-143, 2013.

Three SMC samplers

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

1. Conditional particle filter (CPF)

   Christophe Andrieu, Arnaud Doucet and Roman Holenstein, **Particle Markov chain Monte Carlo methods**, *Journal of the Royal Statistical Society: Series B*, 72:269-342, 2010.

2. CPF with backward simulation (CPF-BS)

   N. Whiteley, **Discussion on Particle Markov chain Monte Carlo methods**, *Journal of the Royal Statistical Society: Series B*, 72(3), 306–307, 2010.

   N. Whiteley, C. Andrieu and A. Doucet, **Efficient Bayesian inference for switching state-space models using discrete particle Markov chain Monte Carlo methods**, *Bristol Statistics Research Report* 10:04, 2010.

   Fredrik Lindsten and Thomas B. Schön. **On the use of backward simulation in the particle Gibbs sampler**. *Proc. of the 37th Internat. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Kyoto, Japan, March 2012.

3. **CPF with ancestor sampling (CPF-AS)**

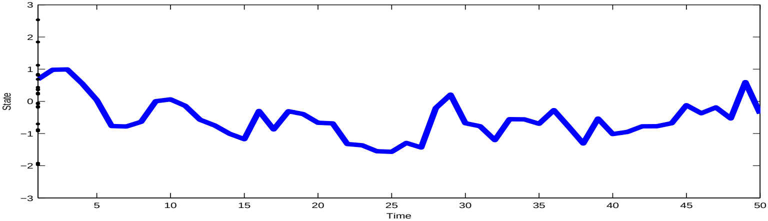
   Fredrik Lindsten, Michael I. Jordan and Thomas B. Schön, **Particle Gibbs with ancestor sampling**, *arXiv:1401.0604*, 2014.

   Fredrik Lindsten, Michael I. Jordan and Thomas B. Schön, **Ancestor sampling for particle Gibbs**, *Advances in Neural Information Processing Systems (NIPS) 25*, Lake Tahoe, NV, US, December, 2012.

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: $x_t^N = x'_t$.



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

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.

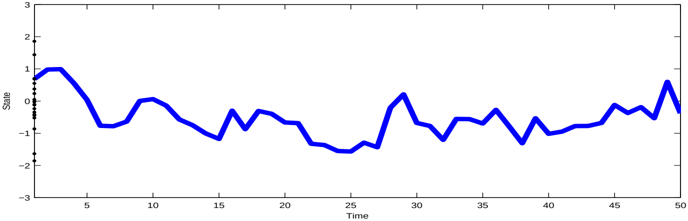
Implication: Ancestor sampling opens up for inference in a wider class of models, e.g. non-Markovian SSMs, PGMs and BNP models.

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 – intuition

Let $x'_{1:T} = (x'_1, \dots, x'_T)$ be a fixed *reference trajectory*.

- At each time t, sample only $N-1$ particles in the standard way.
- Set the N^{th} particle deterministically: $x_t^N = x'_t$.
- Generate an artificial history for x_t^N by ancestor sampling.



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

CPF-AS – algorithm

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

```
1. Initialize ($t = 1$):
    (a) Draw $x_1^i \sim r_{\theta,1}(x_1^i)$, for $i = 1, \dots, N-1$.
    (b) Set $x_1^N = x'_1$.
    (c) Set $w_1^i = W_{\theta,1}(x_1^i)$.
2. For $t = 2$ to $T$ do:
    (a) Draw $a_t^i \sim \text{Cat}(\{w_t^j\}_{j=1}^N)$, for $i = 1, \dots, N-1$.
    (b) Draw $x_t^i \sim r_{\theta,t}(x_t \mid x_{1:t-1}^{a_t^i})$, for $i = 1, \dots, N-1$.
    (c) Set $x_t^N = x'_t$.
    (d) Draw $a_t^N$ with $\mathbb{P}(a_t^N = i) \propto w_{t-1}^i \frac{\gamma_{\theta,T}(x_{1:t-1}^i, x'_{t:T})}{\gamma_{\theta,t-1}(x_{1:t-1}^i)}$.
    (e) Set $x_{1:t}^i = \{x_{1:t-1}^{a_t^i}, x_t^i\}$ and $w_t^i = W_{\theta,t}(x_{1:t}^i)$.
```

The PGAS Markov kernel

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

- Maps $x'_{1:T}$ stochastically into $x^\star_{1:T}$
- Implicitly defines an ergodic Markov kernel (P^N_θ) 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^N_θ *leaves* $p(x_{1:T} \mid \theta, y_{1:T})$ *invariant,*

$$p(dx^\star_{1:T} \mid \theta, y_{1:T}) = \int P^N_\theta(x'_{1:T}, dx^\star_{1:T}) p(dx'_{1:T} \mid \theta, y_{1:T})$$

PGAS for Bayesian learning

Bayesian learning: Gibbs + CPF-AS = PGAS

Algorithm Particle Gibbs with ancestor sampling (PGAS)

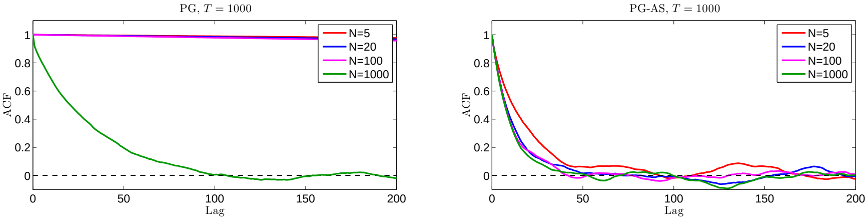
```
1. Initialize: Set $\{\theta[0], x_{1:T}[0]\}$ arbitrarily.
2. For $m \geq 1$, iterate:
    (a) Draw $\theta[m] \sim p(\theta \mid x_{1:T}[m-1], y_{1:T})$.
    (b) Run CPF-AS$(x_{1:T}[m-1])$, targeting $p(x_{1:T} \mid \theta[m], y_{1:T})$.
    (c) Sample with $P(x_{1:T}[m] = x^i_{1:T}) \propto w^i_T$.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

Toy example – stochastic volatility (I/II)

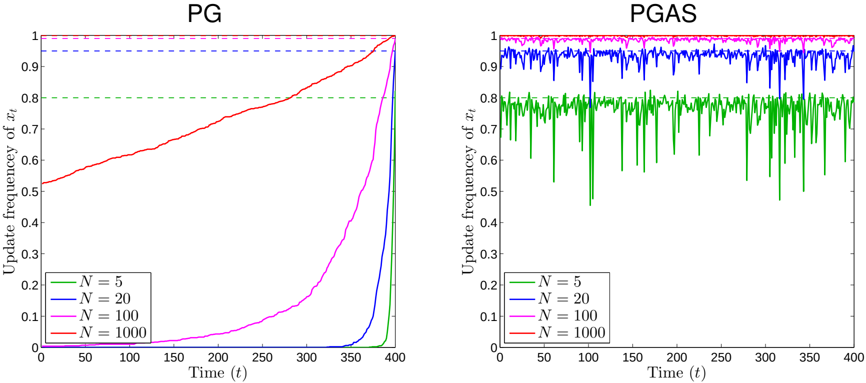
Consider the stochastic volatility model,

$$
\begin{aligned}
x_{t+1} &= 0.9x_t + w_t, & w_t &\sim \mathcal{N}(0, \theta),\\
y_t &= e_t \exp\left(\frac{1}{2}x_t\right), & e_t &\sim \mathcal{N}(0, 1).
\end{aligned}
$$

Let us study the ACF for the estimation error, $\widehat{\theta} - \mathrm{E}\left[\theta \mid y_{1:T}\right]$



Toy example – stochastic volatility (II/II)



Plots of the update rate of x_t versus t, i.e. the proportion of iterations where x_t changes value. This provides another comparison of the mixing.

Two more PGAS application examples

1. A Gaussian process state space model (GP-SSM)

$$\begin{aligned} f(x_t) &\sim \mathcal{GP}(m_{\theta_x}(x_t), k_{\theta_x}(x_t, x_t')), \\ x_{t+1} \mid f_t &\sim \mathcal{N}(x_{t+1} \mid f_t, Q), \\ y_t \mid x_t &\sim p(y_t \mid x_t, \theta_y). \end{aligned}$$

Key idea: Marginalize out the function f using PGAS.

Roger Frigola, Fredrik Lindsten, Thomas B. Schön and Carl E. Rasmussen, **Bayesian inference and learning in Gaussian process state-space models with particle MCMC**. In *Advances in Neural Information Processing Systems (NIPS) 26*, Lake Tahoe, NV, USA, December 2013.

---

2. ML learning using PGAS + stochastic approximation EM (PSAEM)

$$\widehat{Q}_m(\theta) = \widehat{Q}_{m-1}(\theta) + \gamma_m \left(\log p_\theta(x_{1:T}[m], y_{1:T}) - \widehat{Q}_{m-1}(\theta) \right)$$

F. Lindsten, **An efficient stochastic approximation EM algorithm using conditional particle filters**, *Proceedings of the 38th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, Canadan, May 2013.

Outline

1. Maximum Likelihood (ML) learning
   a) Problem formulation
   b) Solution using EM and a particle smoother
2. Bayesian learning
   a) Problem formulation
   b) Gibbs sampling
3. Sequential Monte Carlo (SMC)
4. Particle Gibbs with ancestor sampling (PGAS)
5. **SMC and PGAS for graphical models**

The sequential Monte Carlo samplers are fundamental to **both** the ML and the Bayesian approaches.

Using SMC in graphical models – the idea

Constructing an artificial sequence of intermediate (auxiliary) target distributions for an SMC sampler is a powerful (and **quite possibly underutilized**) idea. For some applications, see

Alexandre Bouchard-Côté and Sriram Sankararaman and Michael I. Jordan. **Phylogenetic Inference via Sequential Monte Carlo**, *Systematic Biology*, 61(4):579–593, 2012.

Pierre Del Moral, Arnaud Doucet and Ajay Jasra. Sequential Monte Carlo samplers, *Journal of the Royal Statistical Society: Series B*, 68(3):411–436, 2006.

Key idea: Perform and make use of a sequential decomposition of the probabilistic graphical model (PGM).

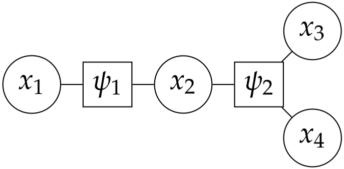
Defines a sequence of intermediate (auxiliary) target distributions defined on an increasing sequence of probability spaces.

Target this sequence using SMC.

Sequential decomposition of PGMs – pictures

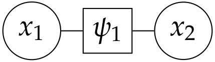
The joint PDF of the set of random variables indexed by $\mathcal{V}$, $X_{\mathcal{V}} \triangleq \{x_1, \ldots, x_{|\mathcal{V}|}\}$

$$p(X_{\mathcal{V}}) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(X_C).$$

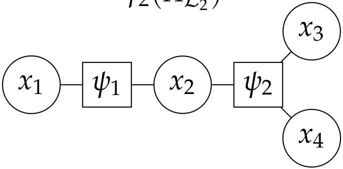


Sequential decomposition of the above factor graph (the target distributions are built up by adding factors at each iteration),

$\gamma_1(X_{\mathcal{L}_1})$



$\gamma_2(X_{\mathcal{L}_2})$



Sequential decomposition of PGMs – equations

Let $\{\psi_k\}_{k=1}^{K}$ be a sequence of factors,

$$\psi_k(X_{\mathcal{I}_k}) = \prod_{C \in \mathcal{C}_k} \psi_C(X_C),$$

where $\mathcal{I}_k \subseteq \{1, \ldots, |\mathcal{V}|\}$.

The **sequential decomposition** is based on these factors,

$$\gamma_k(X_{\mathcal{L}_k}) \triangleq \prod_{\ell=1}^{k} \psi_\ell(X_{\mathcal{I}_\ell}),$$

where $\mathcal{L}_k \triangleq \bigcup_{\ell=1}^{k} \mathcal{I}_\ell$.

By construction, $\mathcal{L}_K = \mathcal{V}$ and the joint PDF $p(X_{\mathcal{L}_K}) \propto \gamma_K(X_{\mathcal{L}_K})$.

SMC sampler for PGMs

Algorithm SMC sampler for PGMs

```
1. Initialize ($k = 1$): Draw $X^i_{\mathcal{L}_1} \sim r_1(\cdot)$ and set $w^i_1 = W_1(X^i_{\mathcal{L}_1})$.
2. For $k = 2$ to $K$ do:
    (a) Draw $a^i_k \sim \text{Cat}(\{w^j_{k-1}\}_{j=1}^N)$.
    (b) Draw $\xi^i_k \sim r_k(\cdot | X^{a^i_k}_{\mathcal{L}_{k-1}})$ and set $X^i_{\mathcal{L}_k} = X^{a^i_k}_{\mathcal{L}_{k-1}} \cup \xi^i_k$.
    (c) Set $w^i_k = W_k(X^i_{\mathcal{L}_k})$.
```

Also provides an estimate of the **partition function**!

Problem: SMC is not enough since:

1. It does not solve the parameter learning problem.
2. The quality of the marginals $p(X_{\mathcal{L}_k})$ deteriorates for $k \ll K$.

Solution: Use PGAS.

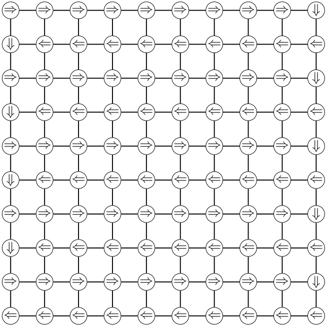
Example – Gaussian MRF (I/II)

Consider a standard squared lattice Gaussian MRF of size 10×10,

$$p(X_{\mathcal{V}}, Y_{\mathcal{V}}) \propto \prod_{i \in \mathcal{V}} e^{\frac{1}{2\sigma_i^2}(x_i - y_i)^2} \prod_{(i,j) \in \mathcal{E}} e^{\frac{1}{2\sigma_{ij}^2}(x_i - x_j)^2}$$

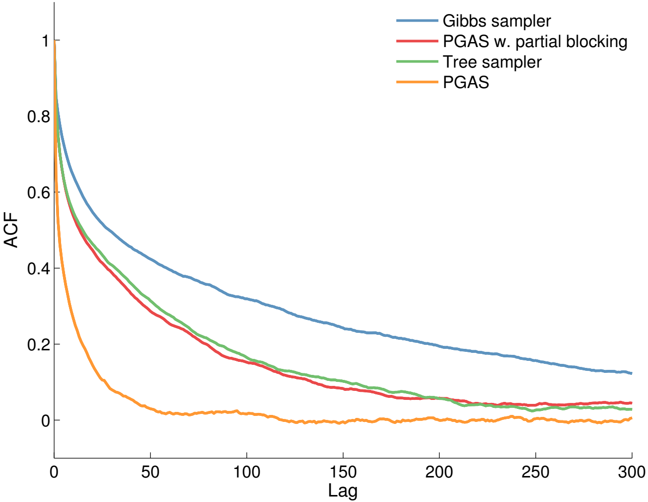
Four MCMC samplers:

1. PGAS – fully blocked
2. PGAS – partially blocked
3. Standard one-at-a-time Gibbs
4. Tree sampler (Hamze & de Freitas, 2004)



The arrows show the order in which the factors are added.

Example – Gaussian MRF (II/II)



Using SMC in graphical models

We have introduced several SMC-based inference methods for PGMs of arbitrary topologies with discrete or continuous variables.

The sequential decomposition is **not unique** and its form will affect

- accuracy
- computational efficiency
- simplicity of implementation

Details and a loopy, non-Gaussian and non-discrete PGM example,

Christian A. Naesseth, Fredrik Lindsten and Thomas B. Schön, **Sequential Monte Carlo methods for graphical models**. *Preprint at arXiv:1402:0330*, February, 2014.

Conclusions

- Think of the PGAS kernel as a **component** that can be used in different inference algorithms.
- **Not** at all limited to SSMs. Particularly useful for models with more complex dependencies, such as
  - Non-Markovian models
  - Bayesian nonparametric models
  - Probabilistic graphical models
- PGAS is built upon two main ideas
  1. **Conditioning** the underlying SMC sampler on a reference trajectory ensures the correct stationary distribution for any N.
  2. **Ancestor sampling** causes degeneration to different trajectories, drastically improving the mixing of the sampler.

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

Some references (I/II)

Novel introduction of PMCMC (given us lots of inspiration)

Christophe Andrieu, Arnaud Doucet and Roman Holenstein, **Particle Markov chain Monte Carlo methods**, *Journal of the Royal Statistical Society: Series B*, 72:269-342, 2010.

Forthcoming book

Thomas B. Schön and Fredrik Lindsten, **Learning of dynamical systems – Particle filters and Markov chain methods**, 2014 (or 2015...).

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

Fredrik Lindsten and Thomas B. Schön, **Backward simulation methods for Monte Carlo statistical inference**, *Foundations and Trends in Machine Learning*, 6(1):1-143, 2013.

PGAS

Fredrik Lindsten, Michael I. Jordan and Thomas B. Schön, **Particle Gibbs with ancestor sampling**, *arXiv:1401.0604*, 2014.

Fredrik Lindsten, Michael I. Jordan and Thomas B. Schön, **Ancestor sampling for particle Gibbs**, *Advances in Neural Information Processing Systems (NIPS) 25*, Lake Tahoe, NV, US, December, 2012.

Some references (II/II)

ML identification of nonlinear SSMs

F. Lindsten, **An efficient stochastic approximation EM algorithm using conditional particle filters**, *Proceedings of the 38th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, Canadan, May 2013.

Thomas B. Schön, Adrian Wills and Brett Ninness. **System Identification of Nonlinear State-Space Models**. *Automatica*, 47(1):39-49, January 2011.

Bayesian inference in SSMs using Gaussian processes

Roger Frigola, Fredrik Lindsten, Thomas B. Schön and Carl E. Rasmussen, **Bayesian inference and learning in Gaussian process state-space models with particle MCMC**. In *Advances in Neural Information Processing Systems (NIPS) 26*, Lake Tahoe, NV, USA, December 2013.

SMC methods for graphical models

Christian A. Naesseth, Fredrik Lindsten and Thomas B. Schön, **Sequential Monte Carlo methods for graphical models**. *Preprint at arXiv:1402:0330*, February, 2014.

Some MATLAB code is available from the web-site.