Sequential MCMC for Bayesian Filtering with Massive Data

François Septier

Institut Mines-Télécom/Télécom Lille/CRIStAL UMR CNRS 9189

Joint work with A. De Freitas and L. Mihaylova (Sheffield Uni., UK)

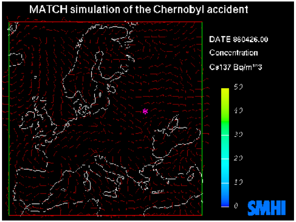
Paper available on arXiv:1512.02452

20-23 July 2016, STM2016 - ISM, Japan

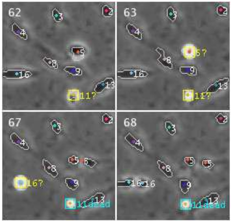
Introduction

In many applications, we are interested in estimating a signal from a sequence of noisy observations.

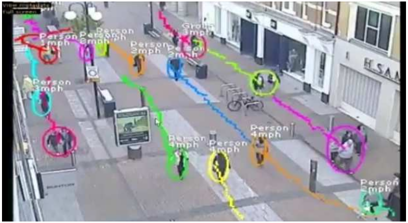
Finance



Environmental monitoring



Computer vision-based
cell tracking algorithms



Video-surveillance

but also in many others...

Introduction : HMM

Such problems are generally formulated by an Hidden Markov Model (HMM) :

- **The hidden State process** : $\{X_n\}_{n\geq 1}$ is a $\mathbb{R}^d$-valued discrete-time Markov process that is not directly observable. The joint distribution of this Markov process $\{X_n\}_{n\geq 1}$ is given by,

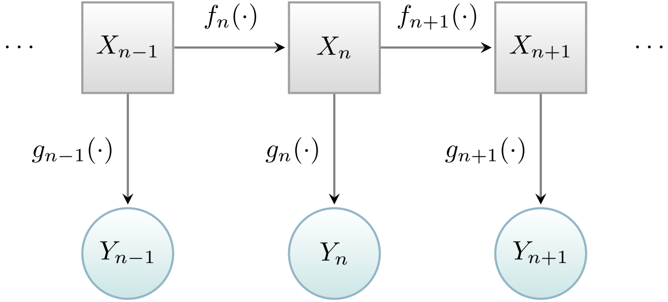
$$p(x_{1:n}) = \mu(x_1)\prod_{k=1}^{n} f_k(x_k|x_{k-1}),$$

- **The observed process** : $\{Y_n\}_{n\geq 1}$ is such that the conditional joint density of $Y_{1:n} = y_{1:n}$ given $X_{1:n} = x_{1:n}$ has the following conditional independence (product) form,

$$p(y_{1:n}|x_{1:n}) = \prod_{k=1}^{n} g_k(y_k|x_k).$$

Introduction : HMM

The HMM can be represented by a graphical model that depicts the conditional independence relations :



The HMM can be considered as the simplest dynamic Bayesian network.

Introduction : Tasks of interest for HMMs

What we generally know :

- the observations $y_{0:k}$
- transition density function $f_k(\cdot|\cdot)$, $\forall k \in \mathbb{N}^+$
- likelihood density function $g_k(\cdot|\cdot)$, $\forall k \in \mathbb{N}^+$

What we want to do :

- **State inference** : How to make probabilistic statements on the state sequence given the model and the observations ?
  Inference about X_n given observations $Y_{1:n} = y_{1:n}$ relies upon the posterior distribution,

$$\pi_n(x_{1:n}) := p(x_{1:n}|y_{1:n}) = \frac{p(x_{1:n}, y_{1:n})}{p(y_{1:n})} = \frac{p(x_{1:n})p(y_{1:n}|x_{1:n})}{p(y_{1:n})}.$$

- **Parameter Inference** How to tune the model parameters based on the observations ?

Filtering recursions

⇒ **Goal :** Estimate sequentially X_n given observations up to time n ($Y_{1:n} = y_{1:n}$)

⇒ The application of Bayes' rule leads to the recursion

$$\underbrace{p(x_{1:n}|y_{1:n})}_{\pi_n(x_{1:n})} = \frac{g_n(y_n|x_n)f_n(x_n|x_{n-1})}{p(y_n|y_{1:n-1})} \underbrace{p(x_{1:n-1}|y_{1:n-1})}_{\pi_{n-1}(x_{1:n-1})},$$

where

$$p(y_n|y_{1:n-1}) = \int g_n(y_n|x_n)f_n(x_n|x_{n-1})p(x_{n-1}|y_{1:n-1})dx_{n-1:n}.$$

Exact implementation of the filtering recursions

⇒ **When $\mathbf{x}$ is finite** (Baum et al., 1970) The associated computational cost is $|x|^2$ per time index (for the filtering part).

⇒ **In linear Gaussian state-space models** (Kalman & Bucy, 1961) The filtering and prediction recursion is implemented by the *Kalman filter*.

However, such exact implementations do not exist for more complex (and thus realistic) models.

Approximate implementation of the filtering recursions

- EKF (Extended Kalman Filter) Linearization-based approach (for non-linear Gaussian state space models)
- UKF (Unscented Kalman Filter) [Julier and Uhlmann, 1997] Point-based approach
- Variational Methods (e.g., [Valpola and Karhunen, 2002]) Based on parametric density approximation arguments.

$\Rightarrow$ These approximations can be seriously unreliable in numerous cases of interest.

Attractive alternatives :
$\rightsquigarrow$ Monte Carlo methods [Handschin and Mayne 1969, Gordon et al., 1993] : they became very popular with the recent availability of high-powered computers.

Traditional MC solution : SMC (particle filter)

Key Idea : Use a sequential version of the *Importance Sampling* algorithm

At each time step k, we do the following steps :

1. Sample independently $X_k^j \sim q_k(\cdot|X_{k-1}^j)$, $\forall j = 1, \cdots, N_p$
2. Compute weight $w_k^j \propto \frac{g_k(y_k|X_k^j) f_k(X_k^j|X_{k-1}^j)}{q_k(X_k^j|X_{k-1}^j)}$, $\forall j = 1, \cdots, N_p$
3. Resample the weighted particle set, $\left\{X_k^j, w_k^j\right\}_{i=1}^{N_p}$, if necessary

Main difficulty : Hard to design an efficient proposal distribution

Are there any (efficient) alternatives to SMC for sequential Bayesian inference ?

$\Rightarrow$ **Use of Markov Chain Monte Carlo (MCMC) in sequential setting.**

Traditional MC solution : SMC (particle filter)

Key Idea : Use a sequential version of the *Importance Sampling* algorithm

At each time step k, we do the following steps :

1. Sample independently $X_k^j \sim q_k(\cdot|X_{k-1}^j)$, $\forall j = 1, \cdots, N_p$
2. Compute weight $w_k^j \propto \frac{g_k(y_k|X_k^j) f_k(X_k^j|X_{k-1}^j)}{q_k(X_k^j|X_{k-1}^j)}$, $\forall j = 1, \cdots, N_p$
3. Resample the weighted particle set, $\left\{X_k^j, w_k^j\right\}_{i=1}^{N_p}$, if necessary

Main difficulty : Hard to design an efficient proposal distribution

Are there any (efficient) alternatives to SMC for sequential Bayesian inference ?

$\Rightarrow$ **Use of Markov Chain Monte Carlo (MCMC) in sequential setting.**

Traditional MC solution : SMC (particle filter)

Key Idea : Use a sequential version of the *Importance Sampling* algorithm

At each time step k, we do the following steps :

1. Sample independently $X_k^j \sim q_k(\cdot|X_{k-1}^j)$, $\forall j = 1, \cdots, N_p$
2. Compute weight $w_k^j \propto \frac{g_k(y_k|X_k^j) f_k(X_k^j|X_{k-1}^j)}{q_k(X_k^j|X_{k-1}^j)}$, $\forall j = 1, \cdots, N_p$
3. Resample the weighted particle set, $\left\{X_k^j, w_k^j\right\}_{i=1}^{N_p}$, if necessary

Main difficulty : Hard to design an efficient proposal distribution

Are there any (efficient) alternatives to SMC for sequential Bayesian inference ?

$\Rightarrow$ **Use of Markov Chain Monte Carlo (MCMC) in sequential setting.**

Alternatives to Importance Sampling based methods $\mapsto$ MCMC :

- $\leadsto$ more effective in high-dimensional and/or complex systems,
- $\leadsto$ more flexible : a lot of different sampling strategies can be used.

Traditionally, MCMC methods $\rightarrow$ Non-sequential setting

But several **Sequential** Markov Chain Monte-Carlo (MCMC) methods exist and have shown promising results !

[Berzuini et al., 1997, Golightly and Wilkinson, 2006, Septier et al., 2009, Brockwell et al., 2010, Septier and Peters, 2016]

Why MCMC methods are generally more effective in complex problems than IS ?

Importance Sampling :

- Difficult to find a suitable proposal distribution in high dimensions

MCMC :

- *Key idea :* Create a dependent sample, i.e. X^n depends on the previous value X^{n-1}.

  $\rightsquigarrow$ allows for "local" updates. ← Key point to deal with high dimensional problems

- *How ?* Construct a Markov chain $X^1, X^2, \ldots$ whose stationary distribution is the target distribution of interest π

Let us briefly recall the principle of MCMC methods

MCMC : Principle

- We know the target distribution up to a normalizing constant : $\pi(x) = \gamma(x)/Z$
- We define a proposal distribution $q(\cdot|x)$
- Initialization of the first sample of the Markov chain X^0
- From the current value of the chain, X^n, we propose a sample from $q(\cdot|X^n)$ and we accept or reject according to some probability that will ensure that the stationary distribution of the Markov chain is the target distribution π
- the first samples of the chain are discarded ("burn-in" period)

Algorithm : Metropolis-Hastings (MH)

Starting with X^0 and iterate for $n = 1, 2, \ldots$

1. Draw $X^* \sim q(\cdot|X^{n-1})$ (Proposal value)
2. Compute

$$\begin{aligned}\alpha(X^*|X^{n-1}) &= \min\left\{1, \frac{\pi(X^*)}{q(X^*|X^{t-1})}\frac{q(X^{n-1}|X^*)}{\pi(X^{n-1})}\right\} \\ &= \min\left\{1, \frac{\gamma(X^*)}{q(X^*|X^{n-1})}\frac{q(X^{n-1}|X^*)}{\gamma(X^{n-1})}\right\}\end{aligned}$$

3. With probability $\alpha(X^*|X^{n-1})$ set $X^n = X^*$, otherwise set $X^n = X^{n-1}$

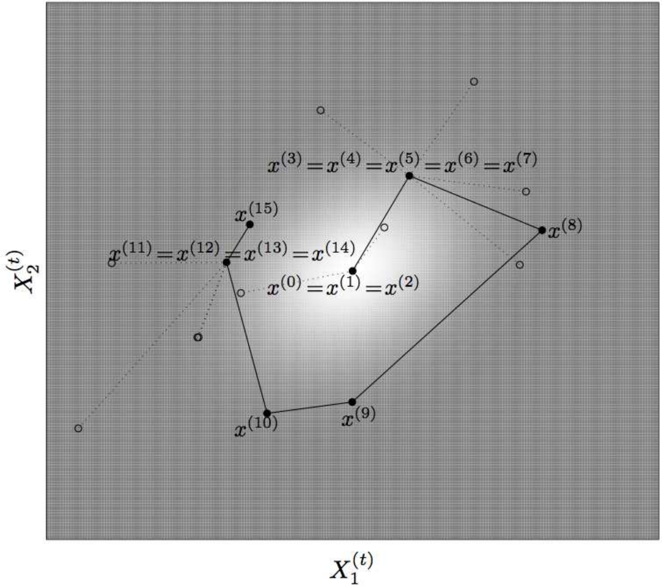


Illustration with a two-dimensional state $(d = 2)$

MCMC : Choice of the MH proposal

- Independent Metropolis-Hastings
  - Take $q(X^*|X^{n-1}) = g(X^*)$ (independent of X^{n-1})
  - g is generally chosen to be an approximation to π
  - Probability of acceptance becomes

$$\min\left\{1, \frac{\gamma(X^*)}{g(X^*)}\frac{g(X^{n-1})}{\gamma(X^{n-1})}\right\}$$

- Random-Walk Metropolis Hastings **[local moves]**
  - The proposal is $q(X^*|X^{n-1}) = g(X^* - X^{n-1})$ with g being a symmetric distribution, thus

$$X^* = X^{n-1} + \epsilon \quad \text{with } \epsilon \sim g$$

  - Probability of acceptance becomes

$$\min\left\{1, \frac{\gamma(X^*)}{g(X^* - X^{n-1})}\frac{g(X^{n-1} - X^*)}{\gamma(X^{n-1})}\right\} = \min\left\{1, \frac{\gamma(X^*)}{\gamma(X^{n-1})}\right\}$$

  - We accept
    - every move to a more probable state with probability 1.
    - moves to less probable states with a probability $\gamma(X^*)/\gamma(X^{n-1}) < 1$

MCMC : Choice of the MH proposal

- Independent Metropolis-Hastings
  - Take $q(X^*|X^{n-1}) = g(X^*)$ (independent of X^{n-1})
  - g is generally chosen to be an approximation to π
  - Probability of acceptance becomes

$$\min\left\{1, \frac{\gamma(X^*)}{g(X^*)} \frac{g(X^{n-1})}{\gamma(X^{n-1})}\right\}$$

- Random-Walk Metropolis Hastings **[local moves]**
  - The proposal is $q(X^*|X^{n-1}) = g(X^* - X^{n-1})$ with g being a symmetric distribution, thus

$$X^* = X^{n-1} + \epsilon \quad \text{with } \epsilon \sim g$$

  - Probability of acceptance becomes

$$\min\left\{1, \frac{\gamma(X^*)}{g(X^* - X^{n-1})} \frac{g(X^{n-1} - X^*)}{\gamma(X^{n-1})}\right\} = \min\left\{1, \frac{\gamma(X^*)}{\gamma(X^{n-1})}\right\}$$

  - We accept
    - every move to a more probable state with probability 1.
    - moves to less probable states with a probability $\gamma(X^*)/\gamma(X^{n-1}) < 1$

Sequential MCMC : General Principle

At time step n, the target distribution of interest to be sampled from is

$$\underbrace{p(x_{1:n}|y_{1:n})}_{\pi_n(x_{1:n})} \propto g_n(y_n|x_n)f_n(x_n|x_{n-1})\underbrace{p(x_{1:n-1}|y_{1:n-1})}_{\pi_{n-1}(x_{1:n-1})}. \tag{1}$$

Impossible to sample from $p(x_{1:n-1}|y_{1:n-1})$ (with constant complexity $\forall n$)

Key Idea of SMCMC :
Replace $p(x_{1:n-1}|y_{1:n-1})$ by an empirical approximation obtained from the algorithm in the previous recursion.

$$\breve{\pi}_n(x_{1:n}) \propto g_n(y_n|x_n)f_n(x_n|x_{n-1})\widehat{\pi}(x_{1:n-1}), \tag{2}$$

with

$$\widehat{\pi}(x_{1:n-1}) = \frac{1}{N}\sum_{m=N_b+1}^{N+N_b} \delta_{X^m_{n-1,1:n-1}}(dx_{1:n-1}), \tag{3}$$

where $\left\{X^m_{n-1,1:n-1}\right\}_{m=N_b+1}^{N+N_b}$: N samples of the Markov chain obtained at the previous $(n-1)$-th time step for which the stationary distribution was $\breve{\pi}_{n-1}(x_{1:n-1})$.

$\Rightarrow$ an MCMC Kernel can thus be employed to obtain a Markov chain $\left(X^1_{n,1:n}, X^2_{n,1:n}, \ldots\right)$, with stationary distribution $\breve{\pi}_n(x_{1:n})$ as defined in Eq. (2).

Sequential MCMC : General Principle

At time step n, the target distribution of interest to be sampled from is

$$\underbrace{p(x_{1:n}|y_{1:n})}_{\pi_n(x_{1:n})} \propto g_n(y_n|x_n) f_n(x_n|x_{n-1}) \underbrace{p(x_{1:n-1}|y_{1:n-1})}_{\pi_{n-1}(x_{1:n-1})}. \quad (1)$$

Impossible to sample from $p(x_{1:n-1}|y_{1:n-1})$ (with constant complexity $\forall n$)

Key Idea of SMCMC :
Replace $p(x_{1:n-1}|y_{1:n-1})$ by an empirical approximation obtained from the algorithm in the previous recursion.

$$\breve{\pi}_n(x_{1:n}) \propto g_n(y_n|x_n) f_n(x_n|x_{n-1}) \widehat{\pi}(x_{1:n-1}), \quad (2)$$

with

$$\widehat{\pi}(x_{1:n-1}) = \frac{1}{N} \sum_{m=N_b+1}^{N+N_b} \delta_{X^m_{n-1,1:n-1}}(dx_{1:n-1}), \quad (3)$$

where $\left\{X^m_{n-1,1:n-1}\right\}_{m=N_b+1}^{N+N_b}$: N samples of the Markov chain obtained at the previous $(n-1)$-th time step for which the stationary distribution was $\breve{\pi}_{n-1}(x_{1:n-1})$.

$\Rightarrow$ an MCMC Kernel can thus be employed to obtain a Markov chain $\left(X^1_{n,1:n}, X^2_{n,1:n}, \dots\right)$, with stationary distribution $\breve{\pi}_n(x_{1:n})$ as defined in Eq. (2).

General SMCMC for filtering

1. If time $n = 1$
2. For $j = 1, \dots, N + N_b$
3. Sample $X^j_{1,1} \sim \mathcal{K}_1(X^{j-1}_{1,1}, \cdot)$ with $\mathcal{K}_1$ an MCMC kernel of invariant distribution $\pi_1(x_1) \propto g_1(y_1|x_1)\mu(x_1)$.
4. Elseif time $n \geq 2$
5. For $j = 1, \dots, N + N_b$
6. *[OPTIONAL]* Refine empirical approximation of previous posterior distributions as described in [Brockwell et al., 2010]
7. Sample $X^j_{n,1:n} \sim \mathcal{K}_n(X^{j-1}_{n,1:n}, \cdot)$ with $\mathcal{K}_n$ an MCMC kernel of invariant distribution $\breve{\pi}_n$ defined in Eq. (2).
8. **Output :** Approximation of the posterior distribution with the following empirical measure :

$$\breve{\pi}_n(x_{1:n}) \approx \frac{1}{N} \sum_{j=N_b+1}^{N+N_b} \delta_{X^j_{n,1:n}}(dx_{1:n})$$

SMCMC : Design of the MCMC Kernel

At each time n the target distribution is

$$\breve{\pi}_n(x_{1:n}) \propto g_n(y_n|x_n) f_n(x_n|x_{n-1}) \sum_{m=N_b+1}^{N+N_b} \delta_{X^m_{n-1,1:n-1}}(dx_{1:n-1}) \tag{4}$$

Empirical posterior $\Rightarrow$ the proposal within the MCMC kernel is such that

$$q(x_{1:n}|X^{i-1}_{n,1:n}) = q(x_n|X^{i-1}_{n,1:n}, x_{1:n-1}) \underbrace{q(x_{1:n-1}|X^{i-1}_{n,1:n})}_{\text{Discrete Support}\left\{X^m_{n-1,1:n-1}\right\}_{m=N_b+1}^{N+N_b}} \tag{5}$$

Sampling from an MCMC kernel of invariant distribution $\breve{\pi}_n$

1. Generate $X^*_{n,1:n-1} \sim \sum_{m=N_b+1}^{N_b+N} \alpha^m \delta_{X^m_{n-1,1:n-1}}(dx_{1:n-1})$
2. Generate $X^*_{n,n} \sim q(x_n|X^{i-1}_{n,1:n}, X^*_{n,1:n-1})$
3. Accept the candidate $X^i_{n,1:n} = X^*_{n,1:n}$ with probability :

$$\alpha = \min\left\{1, \frac{\breve{\pi}_n(X^*_{n,1:n})}{q(X^*_{n,1:n}|X^{i-1}_{n,1:n})} \frac{q(X^{i-1}_{n,1:n}|X^*_{n,1:n})}{\breve{\pi}_n(X^{i-1}_{n,1:n})}\right\}$$

$$= \min\left\{1, \frac{g_n(y_n|X^*_{n,n}) f_n(X^*_{n,n}|X^*_{n,n-1})}{q(X^*_{n,n}|X^{i-1}_{n,1:n}, X^*_{n,1:n-1})\alpha^{m^*}} \frac{q(X^{i-1}_{n,n}|X^*_{n,1:n}, X^{i-1}_{n,1:n-1})\alpha^{m^{i-1}}}{g_n(y_n|X^{i-1}_{n,n}) f_n(X^{i-1}_{n,n}|X^{i-1}_{n,n-1})}\right\}$$

SMCMC : Design of the MCMC Kernel

At each time n the target distribution is

$$\breve{\pi}_n(x_{1:n}) \propto g_n(y_n|x_n) f_n(x_n|x_{n-1}) \sum_{m=N_b+1}^{N+N_b} \delta_{X^m_{n-1,1:n-1}}(dx_{1:n-1}) \qquad (4)$$

Empirical posterior ⇒ the proposal within the MCMC kernel is such that

$$q(x_{1:n}|X^{i-1}_{n,1:n}) = q(x_n|X^{i-1}_{n,1:n}, x_{1:n-1}) \underbrace{q(x_{1:n-1}|X^{i-1}_{n,1:n})}_{\text{Discrete Support}\left\{X^m_{n-1,1:n-1}\right\}_{m=N_b+1}^{N+N_b}} \qquad (5)$$

Sampling from an MCMC kernel of invariant distribution $\breve{\pi}_n$

1. Generate $X^*_{n,1:n-1} \sim \sum_{m=N_b+1}^{N_b+N} \alpha^m \delta_{X^m_{n-1,1:n-1}}(dx_{1:n-1})$
2. Generate $X^*_{n,n} \sim q(x_n|X^{i-1}_{n,1:n}, X^*_{n,1:n-1})$
3. Accept the candidate $X^i_{n,1:n} = X^*_{n,1:n}$ with probability :

$$\alpha = \min\left\{1, \frac{\breve{\pi}_n(X^*_{n,1:n})}{q(X^*_{n,1:n}|X^{i-1}_{n,1:n})} \frac{q(X^{i-1}_{n,1:n}|X^*_{n,1:n})}{\breve{\pi}_n(X^{i-1}_{n,1:n})}\right\}$$

$$= \min\left\{1, \frac{g_n(y_n|X^*_{n,n}) f_n(X^*_{n,n}|X^*_{n,n-1})}{q(X^*_{n,n}|X^{i-1}_{n,1:n}, X^*_{n,1:n-1})\alpha^{m^*}} \frac{q(X^{i-1}_{n,n}|X^*_{n,1:n}, X^{i-1}_{n,1:n-1})\alpha^{m^{i-1}}}{g_n(y_n|X^{i-1}_{n,n}) f_n(X^{i-1}_{n,n}|X^{i-1}_{n,n-1})}\right\}$$

SMCMC : Design of the MCMC Kernel

At each time n the target distribution is

$$\breve{\pi}_n(x_{1:n}) \propto g_n(y_n|x_n) f_n(x_n|x_{n-1}) \sum_{m=N_b+1}^{N+N_b} \delta_{X^m_{n-1,1:n-1}}(dx_{1:n-1}) \qquad (4)$$

Empirical posterior $\Rightarrow$ the proposal within the MCMC kernel is such that

$$q(x_{1:n}|X^{i-1}_{n,1:n}) = q(x_n|X^{i-1}_{n,1:n}, x_{1:n-1}) \underbrace{q(x_{1:n-1}|X^{i-1}_{n,1:n})}_{\text{Discrete Support}\left\{X^m_{n-1,1:n-1}\right\}_{m=N_b+1}^{N+N_b}} \qquad (5)$$

Sampling from an MCMC kernel of invariant distribution $\breve{\pi}_n$

1. Generate $X^*_{n,1:n-1} \sim \sum_{m=N_b+1}^{N_b+N} \alpha^m \delta_{X^m_{n-1,1:n-1}}(dx_{1:n-1})$
2. Generate $X^*_{n,n} \sim q(x_n|X^{i-1}_{n,1:n}, X^*_{n,1:n-1})$
3. Accept the candidate $X^i_{n,1:n} = X^*_{n,1:n}$ with probability :

$$\alpha = \min\left\{1, \frac{\breve{\pi}_n(X^*_{n,1:n})}{q(X^*_{n,1:n}|X^{i-1}_{n,1:n})} \frac{q(X^{i-1}_{n,1:n}|X^*_{n,1:n})}{\breve{\pi}_n(X^{i-1}_{n,1:n})}\right\}$$

$$= \min\left\{1, \frac{g_n(y_n|X^*_{n,n}) f_n(X^*_{n,n}|X^*_{n,n-1})}{q(X^*_{n,n}|X^{i-1}_{n,1:n}, X^*_{n,1:n-1})\alpha^{m^*}} \frac{q(X^{i-1}_{n,n}|X^*_{n,1:n}, X^{i-1}_{n,1:n-1})\alpha^{m^{i-1}}}{g_n(y_n|X^{i-1}_{n,n}) f_n(X^{i-1}_{n,n}|X^{i-1}_{n,n-1})}\right\}$$

Massive data context

At each time $n \rightarrow$ The MCMC kernel requires the computation of the likelihood

$$\alpha = \min \left\{ 1, \frac{g_n(y_n|X^*_{n,n}) f_n(X^*_{n,n}|X^*_{n,n-1})}{q(X^*_{n,n}|X^{i-1}_{n,1:n}, X^*_{n,1:n-1})\alpha^{m^*}} \frac{q(X^{i-1}_{n,n}|X^*_{n,1:n}, X^{i-1}_{n,1:n-1})\alpha^{m^{i-1}}}{g_n(y_n|X^{i-1}_{n,n}) f_n(X^{i-1}_{n,n}|X^{i-1}_{n,n-1})} \right\}$$

$\Rightarrow$ Prohibitive for tall dataset, i.e. y_n contains a large number M_n of individual (independent) data points

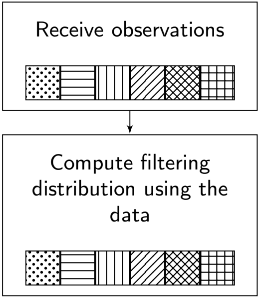
$$g_n(y_n|X^*_{n,n}) = \prod_{k=1}^{M_n} g_n(y_{n,k}|X^*_{n,n})$$

Objective : Adapt recent advances in static MCMC simulation for tall data to the sequential setting.

Techniques for scalable MCMC algorithms can be divided into 2 groups

1. Subsampling-based approaches,
2. Divide-and-Conquer Algorithms

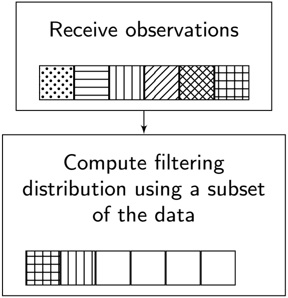
See [Bardenet et al., 2015] for a detailed review.



Techniques for scalable MCMC algorithms can be divided into 2 groups

1. Subsampling-based approaches,
2. Divide-and-Conquer Algorithms

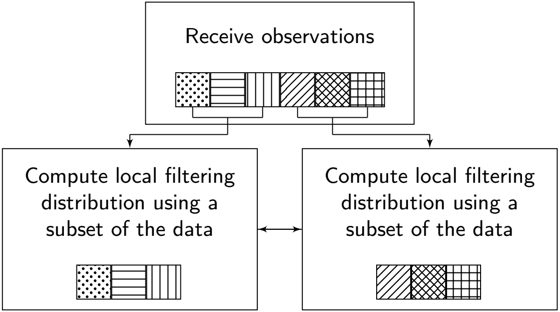
See [Bardenet et al., 2015] for a detailed review.



Techniques for scalable MCMC algorithms can be divided into 2 groups

1. Subsampling-based approaches,
2. Divide-and-Conquer Algorithms

See [Bardenet et al., 2015] for a detailed review.



Subsampling-based approach

Let us recall the acceptance ratio of the SMCMC :

$$\alpha = \min\left\{1, \frac{g_n(y_n|X^*_{n,n})f_n(X^*_{n,n}|X^*_{n,n-1})}{q(X^*_{n,n}|X^{i-1}_{n,1:n}, X^*_{n,1:n-1})\alpha^{m^*}} \frac{q(X^{i-1}_{n,n}|X^*_{n,1:n}, X^{i-1}_{n,1:n-1})\alpha^{m^{i-1}}}{g_n(y_n|X^{i-1}_{n,n})f_n(X^{i-1}_{n,n}|X^{i-1}_{n,n-1})}\right\}$$

The state $X^*_{n,1:n}$ is accepted when (with $u \sim U_{[0,1]}$)

$$u < \frac{\prod_{k=1}^{M_n} g_n(y_{n,k}|X^*_{n,n})f_n(X^*_{n,n}|X^*_{n,n-1})}{q(X^*_{n,n}|X^{i-1}_{n,1:n}, X^*_{n,1:n-1})\alpha^{m^*}} \frac{q(X^{i-1}_{n,n}|X^*_{n,1:n}, X^{i-1}_{n,1:n-1})\alpha^{m^{i-1}}}{\prod_{k=1}^{M_n} g_n(y_{n,k}|X^{i-1}_{n,n})f_n(X^{i-1}_{n,n}|X^{i-1}_{n,n-1})}$$

$$\frac{1}{M_n}\log\left[u\frac{f_n(X^*_{n,n}|X^*_{n,n-1})q(X^{i-1}_{n,n}|X^*_{n,1:n}, X^{i-1}_{n,1:n-1})\alpha^{m^{i-1}}}{f_n(X^{i-1}_{n,n}|X^{i-1}_{n,n-1})q(X^*_{n,n}|X^{i-1}_{n,1:n}, X^*_{n,1:n-1})\alpha^{m^*}}\right] < \frac{1}{M_n}\sum_{k=1}^{M_n}\log\left[\frac{g_n(y_{n,k}|X^*_{n,n})}{g_n(y_{n,k}|X^{i-1}_{n,n})}\right]$$

$$\psi_n(X^*_{n,1:n}, X^{i-1}_{n,1:n}) < \Lambda_{M_n}(X^{i-1}_{n,n}, X^*_{n,n})$$

Subsampling-based approach

[Bardenet et al., 2015] develops a (static) confidence MH sampler for using

$$\Lambda_t^*(X_{n,n}^{i-1}, X_{n,n}^*) = \frac{1}{t}\sum_{k=1}^{t} \log\left[\frac{g_n(y_{n,k}|X_{n,n}^*)}{g_n(y_{n,k}|X_{n,n}^{i-1})}\right]$$

instead of $\Lambda_{M_n}(X_{n,n}^{i-1}, X_{n,n}^*)$ that uses all the data ($t < M_n$)

- By using concentration bounds - for a given $\delta > 0$, $(c_t(\delta), t)$ can be found such that

$$\mathbb{P}\left[|\Lambda_t^*(X_{n,n}^{i-1}, X_{n,n}^*) - \Lambda_{M_n}(X_{n,n}^{i-1}, X_{n,n}^*)| \leq c_t(\delta)\right] \geq 1 - \delta$$

  $\rightsquigarrow$ sampling t from M_n data points without replacement

$$c_t(\delta) = \hat{\sigma}_t\sqrt{\frac{2\log(3/\delta)}{t}} + \frac{3R\log(3/\delta)}{t} \quad \text{[Empirical Berstein Bound]}$$

  with $\hat{\sigma}_t$: empirical std of the log likelihood ratios.
  $R = \max_{1\leq k\leq M_n} |\log g_n(y_{n,k}|X_{n,n}^*) - \log g_n(y_{n,k}|X_{n,n}^{i-1})|$

- Propose an adaptive procedure for t such that the MH acceptance decision is recovered with probability $1 - \delta$
  increase t until the condition
  $|\Lambda_t^*(X_{n,n}^{i-1}, X_{n,n}^*) - \psi_n(X_{n,1:n}^*, X_{n,1:n}^{i-1})| > c_t(\delta)$ is satisfied

Subsampling-based approach

[Bardenet et al., 2015] develops a (static) confidence MH sampler for using

$$\Lambda_t^*(X_{n,n}^{i-1}, X_{n,n}^*) = \frac{1}{t}\sum_{k=1}^{t} \log\left[\frac{g_n(y_{n,k}|X_{n,n}^*)}{g_n(y_{n,k}|X_{n,n}^{i-1})}\right]$$

instead of $\Lambda_{M_n}(X_{n,n}^{i-1}, X_{n,n}^*)$ that uses all the data ($t < M_n$)

- By using concentration bounds - for a given $\delta > 0$, $(c_t(\delta), t)$ can be found such that

  $$\mathbb{P}\left[|\Lambda_t^*(X_{n,n}^{i-1}, X_{n,n}^*) - \Lambda_{M_n}(X_{n,n}^{i-1}, X_{n,n}^*)| \leq c_t(\delta)\right] \geq 1-\delta$$

  $\rightsquigarrow$ sampling t from M_n data points without replacement

  $$c_t(\delta) = \hat{\sigma}_t\sqrt{\frac{2\log(3/\delta)}{t}} + \frac{3R\log(3/\delta)}{t} \quad \text{[Empirical Berstein Bound]}$$

  with $\hat{\sigma}_t$: empirical std of the log likelihood ratios.
  $R = \max_{1\leq k\leq M_n} |\log g_n(y_{n,k}|X_{n,n}^*) - \log g_n(y_{n,k}|X_{n,n}^{i-1})|$

- Propose an adaptive procedure for t such that the MH acceptance decision is recovered with probability $1-\delta$
  increase t until the condition
  $|\Lambda_t^*(X_{n,n}^{i-1}, X_{n,n}^*) - \psi_n(X_{n,1:n}^*, X_{n,1:n}^{i-1})| > c_t(\delta)$ is satisfied

Subsampling-based approach

[Bardenet et al., 2015] develops a (static) confidence MH sampler for using

$$\Lambda_t^*(X_{n,n}^{i-1}, X_{n,n}^*) = \frac{1}{t}\sum_{k=1}^{t} \log\left[\frac{g_n(y_{n,k}|X_{n,n}^*)}{g_n(y_{n,k}|X_{n,n}^{i-1})}\right]$$

instead of $\Lambda_{M_n}(X_{n,n}^{i-1}, X_{n,n}^*)$ that uses all the data ($t < M_n$)

- By using concentration bounds - for a given $\delta > 0$, $(c_t(\delta), t)$ can be found such that

  $$\mathbb{P}\left[|\Lambda_t^*(X_{n,n}^{i-1}, X_{n,n}^*) - \Lambda_{M_n}(X_{n,n}^{i-1}, X_{n,n}^*)| \leq c_t(\delta)\right] \geq 1 - \delta$$

  $\rightsquigarrow$ sampling t from M_n data points without replacement

  $$c_t(\delta) = \hat{\sigma}_t\sqrt{\frac{2\log(3/\delta)}{t}} + \frac{3R\log(3/\delta)}{t} \quad \text{[Empirical Berstein Bound]}$$

  with $\hat{\sigma}_t$: empirical std of the log likelihood ratios.
  $R = \max_{1\leq k\leq M_n} |\log g_n(y_{n,k}|X_{n,n}^*) - \log g_n(y_{n,k}|X_{n,n}^{i-1})|$
- Propose an adaptive procedure for t such that the MH acceptance decision is recovered with probability $1 - \delta$
  increase t until the condition
  $|\Lambda_t^*(X_{n,n}^{i-1}, X_{n,n}^*) - \psi_n(X_{n,1:n}^*, X_{n,1:n}^{i-1})| > c_t(\delta)$ is satisfied

In the empirical Bernstein bound,

$$c_t(\delta) = \hat{\sigma}_t \sqrt{\frac{2\log(3/\delta)}{t}} + \frac{3R\log(3/\delta)}{t} \quad \text{[Empirical Berstein Bound]}$$

the leading term is $\hat{\sigma}_t/\sqrt{t}$
where $\hat{\sigma}_t$: empirical std of the log likelihood ratios

$$\left\{\log \frac{g_n(y_{n,k}|X^*_{n,n})}{g_n(y_{n,k}|X^{i-1}_{n,n})}, k = 1, \ldots, t\right\}$$

To reduce this term, [Bardenet et al., 2015] proposes to use proxies as control variates

Subsampling-based approach with control variates

Assume you have

$$\wp_{n,k}(X_{n,n}^{i-1}, X_{n,n}^{*}) \approx \log g_n(y_{n,k}|X_{n,n}^{*}) - \log g_n(y_{n,k}|X_{n,n}^{i-1})$$

then the MH acceptance decision is equivalent to

$$\frac{1}{M_n}\sum_{k=1}^{M_n}\left[\log\frac{g_n(y_{n,k}|X_{n,n}^{*})}{g_n(y_{n,k}|X_{n,n}^{i-1})} - \wp_{n,k}(X_{n,n}^{i-1}, X_{n,n}^{*})\right] > \psi_n(X_{n,1:n}^{*}, X_{n,1:n}^{i-1}) - \frac{1}{M_n}\sum_{k=1}^{M_n}\wp_{n,k}(X_{n,n}^{i-1}, X_{n,n}^{*})$$

and the leading term of Bernstein's bound now uses the std of

$$\left\{\log\frac{g_n(y_{n,k}|X_{n,n}^{*})}{g_n(y_{n,k}|X_{n,n}^{i-1})} - \wp_{n,k}(X_{n,n}^{i-1}, X_{n,n}^{*}), k = 1, \ldots, t\right\}$$

Example of proxy ⇝ Taylor series of the log-likelihood ratio

- Average of the proxies $\frac{1}{M_n}\sum_{k=1}^{M_n}\wp_{n,k}(X_{n,n}^{i-1}, X_{n,n}^{*})$ easy to compute
- Bound $R = \max_{1\le k\le M_n}\left|\log\frac{g_n(y_{n,k}|X_{n,n}^{*})}{g_n(y_{n,k}|X_{n,n}^{i-1})} - \wp_{n,k}(X_{n,n}^{i-1}, X_{n,n}^{*})\right|$ obtained from the Taylor-Lagrange inequality

Assume you have

$$\wp_{n,k}(X_{n,n}^{i-1}, X_{n,n}^{*}) \approx \log g_n(y_{n,k}|X_{n,n}^{*}) - \log g_n(y_{n,k}|X_{n,n}^{i-1})$$

then the MH acceptance decision is equivalent to

$$\frac{1}{M_n}\sum_{k=1}^{M_n}\left[\log\frac{g_n(y_{n,k}|X_{n,n}^{*})}{g_n(y_{n,k}|X_{n,n}^{i-1})} - \wp_{n,k}(X_{n,n}^{i-1}, X_{n,n}^{*})\right] > \psi_n(X_{n,1:n}^{*}, X_{n,1:n}^{i-1}) - \frac{1}{M_n}\sum_{k=1}^{M_n}\wp_{n,k}(X_{n,n}^{i-1}, X_{n,n}^{*})$$

and the leading term of Bernstein's bound now uses the std of

$$\left\{\log\frac{g_n(y_{n,k}|X_{n,n}^{*})}{g_n(y_{n,k}|X_{n,n}^{i-1})} - \wp_{n,k}(X_{n,n}^{i-1}, X_{n,n}^{*}), k = 1,\ldots,t\right\}$$

Example of proxy $\rightsquigarrow$ Taylor series of the log-likelihood ratio

- Average of the proxies $\frac{1}{M_n}\sum_{k=1}^{M_n}\wp_{n,k}(X_{n,n}^{i-1}, X_{n,n}^{*})$ easy to compute
- Bound $R = \max_{1\le k\le M_n}\left|\log\frac{g_n(y_{n,k}|X_{n,n}^{*})}{g_n(y_{n,k}|X_{n,n}^{i-1})} - \wp_{n,k}(X_{n,n}^{i-1}, X_{n,n}^{*})\right|$ obtained from the Taylor-Lagrange inequality

Assume you have

$$\wp_{n,k}(X_{n,n}^{i-1}, X_{n,n}^*) \approx \log g_n(y_{n,k}|X_{n,n}^*) - \log g_n(y_{n,k}|X_{n,n}^{i-1})$$

then the MH acceptance decision is equivalent to

$$\frac{1}{M_n}\sum_{k=1}^{M_n}\left[\log\frac{g_n(y_{n,k}|X_{n,n}^*)}{g_n(y_{n,k}|X_{n,n}^{i-1})} - \wp_{n,k}(X_{n,n}^{i-1}, X_{n,n}^*)\right] > \psi_n(X_{n,1:n}^*, X_{n,1:n}^{i-1}) - \frac{1}{M_n}\sum_{k=1}^{M_n}\wp_{n,k}(X_{n,n}^{i-1}, X_{n,n}^*)$$

and the leading term of Bernstein's bound now uses the std of

$$\left\{\log\frac{g_n(y_{n,k}|X_{n,n}^*)}{g_n(y_{n,k}|X_{n,n}^{i-1})} - \wp_{n,k}(X_{n,n}^{i-1}, X_{n,n}^*), k = 1, \ldots, t\right\}$$

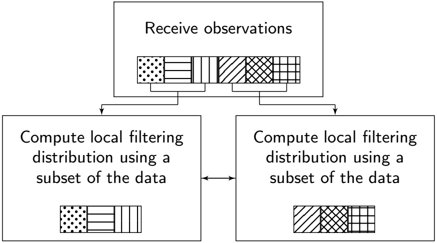
Example of proxy $\rightsquigarrow$ Taylor series of the log-likelihood ratio

- Average of the proxies $\frac{1}{M_n}\sum_{k=1}^{M_n}\wp_{n,k}(X_{n,n}^{i-1}, X_{n,n}^*)$ easy to compute
- Bound $R = \max_{1\le k\le M_n}|\log\frac{g_n(y_{n,k}|X_{n,n}^*)}{g_n(y_{n,k}|X_{n,n}^{i-1})} - \wp_{n,k}(X_{n,n}^{i-1}, X_{n,n}^*)|$ obtained from the Taylor-Lagrange inequality

Divide-and-Conquer based approach

Previous approach : Subsampling ⇝ only a subset of all the data is used

Now we adapt (in the sequential setting) a divide-and-conquer approach based on Expectation-Propagation (EP) [Xu et al., 2014, Gelman et al., 2014]



Key Idea :

1. Partition the M_n measurements into D (disjoint) subsets
2. Run a filter locally on each subset

Challenge : How to combine results from local computation
⇝ EP (variational message passing algorithm) [Minka, 2001]

Divide-and-Conquer - EP SMCMC

Let us recall the true target distribution

$$\breve{\pi}_n(x_n) \propto \prod_{d=1}^{D} g_n(y_{n,\Omega_d}|x_n) \sum_{m=N_b+1}^{N+N_b} f_n(x_n|x_{n-1} = X^m_{n-1,1:n-1})$$

We define a **local** target distribution for an individual computing node :

$$\breve{\pi}^d_n(x_n) \propto g_n(y_{n,\Omega_d}|x_n) \prod_{\substack{c=1 \\ \neq d}}^{D} h(x_n;\eta_c) \sum_{m=N_b+1}^{N+N_b} f_n(x_n|x_{n-1} = X^m_{n-1,1:n-1})$$

where the distribution $h(x_n;\eta_c)$ (e.g. from an exponential family with natural parameters η_c) is an approximation of the likelihood on the c-th node.

Divide-and-Conquer - EP SMCMC

At the dth note, the **local** target distribution is :

$$\breve{\pi}_n^d(x_n) \propto g_n(y_{n,\Omega_d}|x_n) \prod_{\substack{c=1 \\ \neq d}}^{D} h(x_n;\eta_c) \sum_{m=N_b+1}^{N+N_b} f_n(x_n|x_{n-1} = X_{n-1,1:n-1}^m)$$

1. Draw samples from the MCMC kernel with invariant distribution $\breve{\pi}_n^d(x_n)$
2. Update the natural parameters (NP), η_d associated to the likelihood used in this node $\rightsquigarrow$ KL minimization which leads to

$$\eta_d = \eta_{p,d} - \left(\eta_{f,d} + \sum_{i \neq d} \eta_i\right)$$

3. These natural parameters are distributed to all $D \setminus d$ computing nodes.

This procedure is

- performed on all nodes which distribute their NP update to the other ones
- repeated several times.

Finally, the samples from all the local nodes (of the last EP iter.) are kept for approximating of the posterior distribution.

Divide-and-Conquer - EP SMCMC

At the dth note, the **local** target distribution is :

$$\breve{\pi}_n^d(x_n) \propto g_n(y_{n,\Omega_d}|x_n) \prod_{\substack{c=1\\ \neq d}}^{D} h(x_n;\eta_c) \sum_{m=N_b+1}^{N+N_b} f_n(x_n|x_{n-1} = X_{n-1,1:n-1}^m)$$

1. Draw samples from the MCMC kernel with invariant distribution $\breve{\pi}_n^d(x_n)$
2. Update the natural parameters (NP), η_d associated to the likelihood used in this node $\rightsquigarrow$ KL minimization which leads to

$$\eta_d = \eta_{p,d} - \left(\eta_{f,d} + \sum_{i\neq d} \eta_i\right)$$

3. These natural parameters are distributed to all $D \setminus d$ computing nodes.

This procedure is

- performed on all nodes which distribute their NP update to the other ones
- repeated several times.

Finally, the samples from all the local nodes (of the last EP iter.) are kept for approximating of the posterior distribution.

Divide-and-Conquer - EP SMCMC

At the dth note, the **local** target distribution is :

$$\breve{\pi}_n^d(x_n) \propto g_n(y_{n,\Omega_d}|x_n) \prod_{\substack{c=1\\ \neq d}}^{D} h(x_n;\eta_c) \sum_{m=N_b+1}^{N+N_b} f_n(x_n|x_{n-1} = X_{n-1,1:n-1}^m)$$

1. Draw samples from the MCMC kernel with invariant distribution $\breve{\pi}_n^d(x_n)$
2. Update the natural parameters (NP), η_d associated to the likelihood used in this node $\rightsquigarrow$ KL minimization which leads to

$$\eta_d = \eta_{p,d} - \left(\eta_{f,d} + \sum_{i \neq d} \eta_i\right)$$

3. These natural parameters are distributed to all $D \setminus d$ computing nodes.

This procedure is

- performed on all nodes which distribute their NP update to the other ones
- repeated several times.

Finally, the samples from all the local nodes (of the last EP iter.) are kept for approximating of the posterior distribution.

Numerical Simulations

We compare performances of :

- SMCMC : Sequential MCMC
- AS-SMCMC : Adaptive Subsampling SMCMC
  $\rightsquigarrow$ 2nd order Taylor series of log lik. as proxy
- EP-SMCMC : Expectation-Propagation SMCMC
  $\rightsquigarrow$ Multivariate normal distribution for local approx.

in two differents models

- linear and Gaussian state-space model,
- Multiple target tracking in clutter.

Numerical Simulations

We compare performances of :

- SMCMC : Sequential MCMC
- AS-SMCMC : Adaptive Subsampling SMCMC
  $\rightsquigarrow$ 2nd order Taylor series of log lik. as proxy
- EP-SMCMC : Expectation-Propagation SMCMC
  $\rightsquigarrow$ Multivariate normal distribution for local approx.

in two differents models

- linear and Gaussian state-space model,
- Multiple target tracking in clutter.

Numerical Simulations : Model 1

$$f_n(x_n|x_{n-1}) = \mathcal{N}(x_n; Ax_{n-1}, Q)$$
$$g_n(y_n|x_n) = \prod_{k=1}^{M_n} g_n(y_{n,k}|x_n) = \prod_{k=1}^{M_n} \mathcal{N}(y_{n,k}; Hx_k, R)\,.$$

Within this model, the filtering distribution is tractable $\rightsquigarrow$ Kalman filter

Parameters of the different algorithms chosen such that the number of generated samples is the same.

Numerical Simulations : Model 1

Table – Algorithm computation time per time step (AS-SMCMC/SMCMC : $N_p = 4000$ - EP-SMCMC : $L = 2,\ D = 4$ and $N_p = 500$.

Algorithms	$M_n = 500$		$M_n = 5000$	
	Time [s]	Computational Gain [%]	Time [s]	Computational Gain [%]
SMCMC	114.75	0	1087.93	0
AS-SMCMC	69.54	39.4	274.60	74.76
EP-SMCMC	9.89	91.38	96.40	91.14

$\Rightarrow$ Computational saving with both AS and EP

Numerical Simulations : Model 1

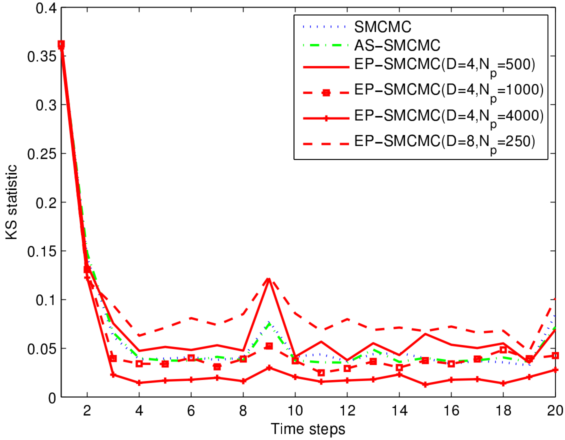
To analyze the quality of the empirical approx. of the filtering distribution :
$\rightsquigarrow$ Study of the Kolmogorov-Smirnov (KS) statistic

$$KS = \sup_x \left(\widehat{F}(x) - G(x) \right),$$

where

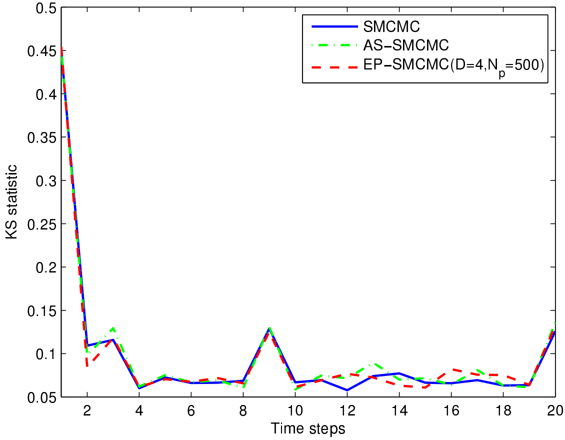
- $\widehat{F}(x)$: empirical cumulative density function of the filtering obtained from the MCMC samples
- $G(x)$: true filtering cdf from the Kalman filter.

Figure – KS statistics with 500 measurements



- Quite similar performances for SMCMC and AS-SMCMC $(1 - \delta = 90\%)$
- EP-SMCMC depends on #nodes (D) and #particles per node N_p
  ⇝ Favorable scenario for EP-SMCMC since Gaussian is used as approx.

Figure – KS statistics with 5000 measurements



Numerical Simulations : MTT

Aim : Detect, track and identify each targets from a sequence of noisy observations.
State-space model :

- Each target follows independently some dynamical model (e.g. near constant velocity model)
- Observation Model : Poisson point process model [Gilholm and Salmond, 2005]

Assumed a set of sensor measurements $y_n = \{y_{n,1}, ..., y_{n,M_n}\}$ coming from a target or clutter (false alarm).
The likelihood function of the observations can be expressed as

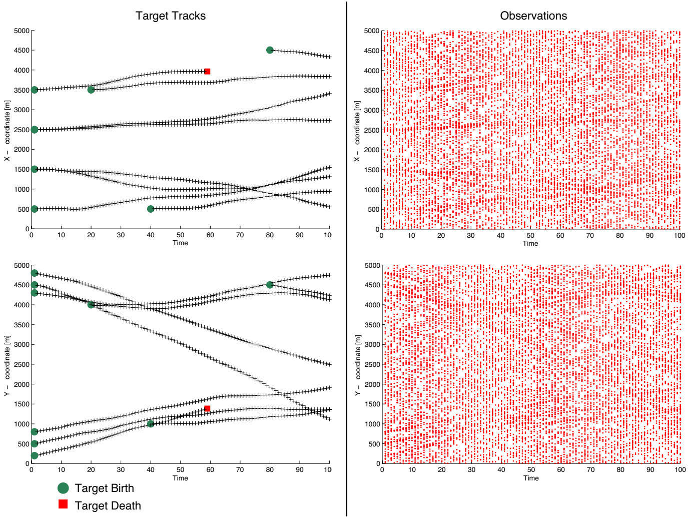
$$g_n(y_n|x_n) = \frac{e^{-\mu_n}}{M_n!} \prod_{m=1}^{M_n} \lambda(y_{n,m})$$

where $\mu_n = \Lambda_C + N_{T,n}\Lambda_x^n$ is the expected total number of measurements received at time t_n and

$$\lambda(y_{n,m}) = \sum_{k=1}^{N_{T,n}} \Lambda_x^n p_x(y_{n,m}|x_{n,k}) + \Lambda_C p_C(y_{n,m})$$

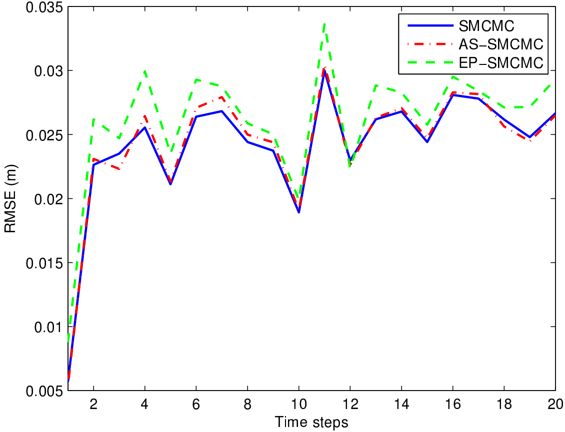
with $\Lambda_x^n p_x(.)$ and $\Lambda_C p_C(.)$ being the Poisson intensity functions of target and clutter measurements and $N_{T,n}$ the number of targets at time t_n.

Figure – Exemple of target' trajectory and associated measurements



Numerical Simulations : MTT

Figure – Root Mean Square Error on the targets' position



Good tracking performances but

- some RMSE increase for the EP-SMCMC $\rightsquigarrow$ Gaussian Approx. likelihood.

Conclusion

- Adapt to the sequential setting two recent approaches proposed for static MCMC with tall dataset
- Interesting computational savings,
- Expectation-Propagation based algo suffers from the choice of parametric distribution to use to approximate local likelihoods

Ongoing work :

- Study the non uniform sampling with replacement in the Adative Subsampling approach.

Bibliography

Bardenet, R., Doucet, A. and Holmes, C. (2015).
On Markov chain Monte Carlo methods for tall data.
arXiv.org , 1–42.

Berzuini, C., Best, N. G., Gilks, W. R. and Larizza, C. (1997).
Dynamic Conditional Independence Models and Markov Chain Monte Carlo Methods.
J. Am. Stat. Assoc. *92*, 1403–1412.

Brockwell, A., Del Moral, P. and Doucet, A. (2010).
Sequentially interacting Markov chain Monte Carlo methods.
Ann. Stat. *38*, 3387–3411.

Gelman, A., Vehtari, A., Jylänki, P., Robert, C., Chopin, N. and Cunningham, J. P. (2014).
Expectation propagation as a way of life.
arXiv.org , 1–29.

Gilholm, K. and Salmond, D. (2005).
Spatial distribution model for tracking extended objects.
IEE Proceedings - Radar, Sonar and Navigation *152*, 364.

Golightly, A. and Wilkinson, D. (2006).
Bayesian sequential inference for nonlinear multivariate diffusions.
Stat. and Comput. *16*, 323–338.

Minka, T. P. (2001).
A family of algorithms for approximate Bayesian inference.
PhD thesis, Massachusetts Institute of Technology.

Septier, F., Pang, S., Carmi, A. and Godsill, S. (2009).
On MCMC-Based Particle Methods for Bayesian Filtering : Application to Multitarget Tracking.
In Proc. IEEE Int. Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), Aruba, Dutch Antilles.

Septier, F. and Peters, G. W. (2016).
Langevin and Hamiltonian Based Sequential MCMC for Efficient Bayesian Filtering in High-Dimensional Spaces.
IEEE Journal of Selected Topics in Signal Processing *10*, 312–327.

Xu, M., Teh, Y. W., Zhu, J. and Zhang, B. (2014).
Distributed Context-Aware Bayesian Posterior Sampling via Expectation Propagation.
In Advances in Neural Information Processing Systems, **Montreal**.