Langevin and Hamiltonian based Sequential MCMC for Efficient Bayesian Filtering in High-dimensional Spaces

François Septier

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

Joint work with Gareth W. Peters (UCL, UK)

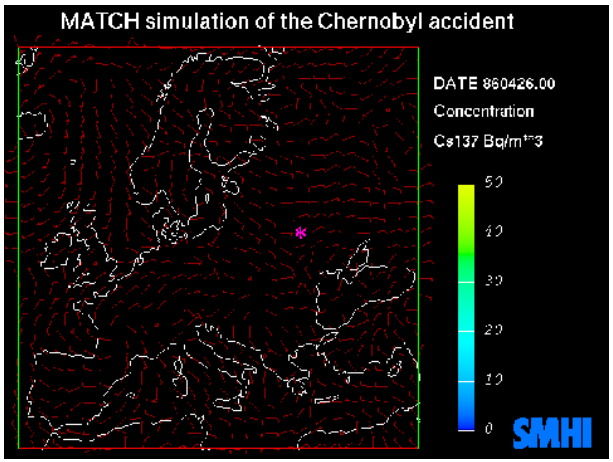
Paper available on arXiv:1504.05715

July 16th 2015, STM/CSM 2015 - Japan

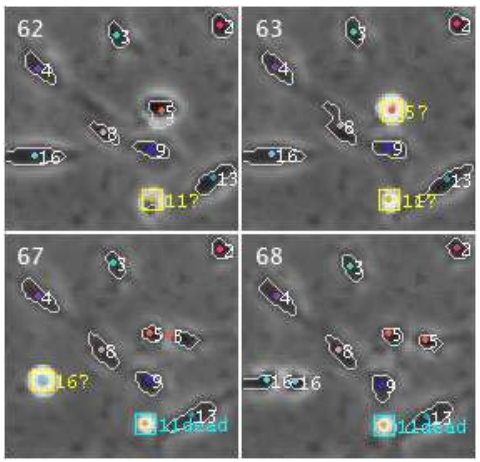
Introduction

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

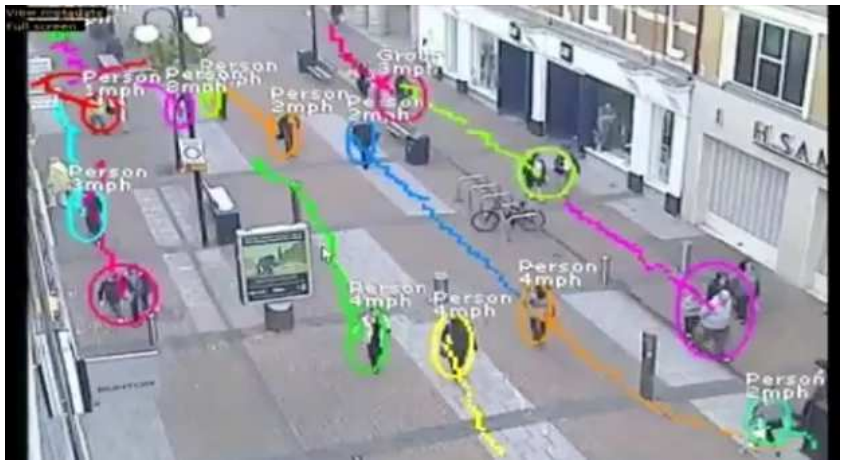
Finance



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,

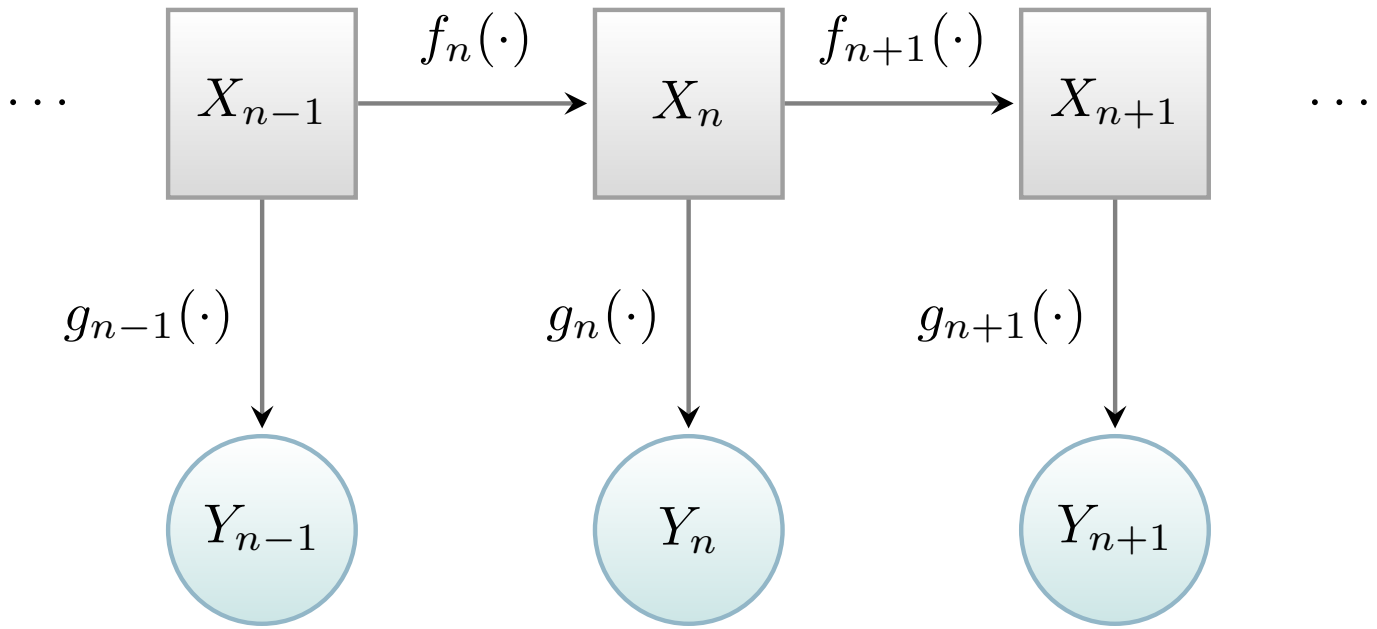
$$p(x_{1:n}) = \mu(x_1) \prod_{k=1}^{n} f_k(x_k|x_{k-1}),$$

- **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

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

$\Rightarrow$ 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

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

$\Rightarrow$ **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 & Uhlmann, 1997) Point-based approach
- Variational Methods (e.g., Valpola & 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 & Mayne 1969, Gordon et al., 1993) : they became very popular with the recent availability of high-powered computers.

Outline

Importance Sampling

Let us define the target distribution of interest which is known up to a normalizing constant Z :

$$\pi(x) = \frac{\gamma(x)}{Z}$$

Importance Sampling (IS) identity

For any distribution q such that $\text{supp}(\pi) \subset \text{supp}(q)$

$$\mathbb{E}_\pi[h(X)] = \int h(x)\frac{\pi(x)}{q(x)}q(x)dx$$

$q(\cdot)$ is called *importance (or proposal / instrumental) distribution*
$w(x) = \pi(x)/q(x)$ is called *importance weight*.

$q(\cdot)$ can be chosen arbitrarily, in particular easy to sample from.

Importance Sampling

Target distribution of interest : $\pi(x) = \frac{\gamma(x)}{Z}$

Importance Sampling (IS) identity

For any distribution q such that $\text{supp}(\pi) \subset \text{supp}(q)$

$$\mathbb{E}_\pi[h(X)] = \int h(x)w(x)q(x)dx \quad \text{with } w(x) = \pi(x)/q(x)$$

- Draw independently N_p samples from $q(\cdot)$

$$\text{for } j = 1, \ldots, N_p : \; X^j \overset{iid}{\sim} q(\cdot)$$

- Plugging this expression in the IS identity, we obtain [by the Law of Large numbers] :

$$\frac{1}{N}\sum_{i=1}^{N} w(X^i)h(X^i) \overset{a.s.}{\underset{N \to \infty}{\longrightarrow}} \mathbb{E}_\pi[h(X)]$$

or self-normalized version (used when Z is unknown)

$$\sum_{i=1}^{N} \frac{w(X^i)}{\sum_{j=1}^{N} w(X^j)} h(X^i) \overset{a.s.}{\underset{N \to \infty}{\longrightarrow}} \mathbb{E}_\pi[h(X)]$$

$$\text{with } w(X^i) = \frac{\gamma(X^i)}{q(X^i)}$$

Back to the filtering problem

Aim : Approximate the filtering distribution $\{\pi_k(x_k) = p(x_k|y_{0:k})\}_{k\geq 0}$ sequentially as the observations are received

How can we use Importance Sampling to do it sequentially (with constant computational load at each time step) ?

$\Rightarrow$ Use the smoothing recursion

$$\begin{aligned}\pi_{k+1}(x_{1:k+1}) &= p(x_{1:k+1}|y_{1:k+1}) \\ &= \underbrace{\left(\frac{p(y_{1:k+1})}{p(y_{1:k})}\right)^{-1}}_{Z_{k+1}^{-1}} \underbrace{\pi_k(x_{1:k}) f_{k+1}(x_{k+1}|x_k) g_{k+1}(y_{k+1}|x_{k+1})}_{\gamma_{k+1}(x_{1:k+1})}\end{aligned}$$

Note : The normalizing constant Z_k is generally unknown

Sequential Importance Sampling

- At initial time, we use IS with $q_1(\cdot)$ as proposal to obtain an approximation of $\pi_1(x_1)$:

$$\widehat{\pi}_1(x_1) = \sum_{i=1}^{N_p} \frac{w_1^i}{\sum_{j=1}^{N_p} w_1^j} \delta_{x_1^i}(dx_0) \text{ with } w_1^i = \frac{\gamma_1(X_1^i)}{q_1(X_1^i)} = \frac{\mu(X_1^i) g_1(y_1|X_1^i)}{q_1(X_1^i)}$$

- At time 2, we want to approximate $\pi_2(x_{1:2})$ by re-using the samples previously obtained $\left\{X_1^i\right\}_{i=1}^{N_p}$ with proposal $q_0(\cdot)$

$\Rightarrow$ Possible by selecting the proposal at time 2 :

$$q_2(x_{1:2}) = q_2(x_2|x_1) q_1(x_1)$$

so $\left\{X_{1:2}^i\right\} \sim q_2(x_{0:1})$ by just adding $\left\{X_2^i\right\} \sim q_2(x_2|X_1^i)$ to all previous particle trajectories ($i = 1, \ldots, N_p$).

Sequential Importance Sampling

- Thus, the computation of the importance weights is :

$$\begin{aligned} w_2(x_{1:2}) &= \frac{\gamma_2(x_{1:2})}{q_2(x_{1:2})} = \frac{\gamma_2(x_{1:2})}{q_2(x_2|x_1)q_1(x_1)} \\ &= \frac{\gamma_1(x_1)}{q_1(x_1)} \frac{\gamma_2(x_{1:2})}{\gamma_1(x_1)q_2(x_2|\mathbf{x}_1)} \\ &= \underbrace{w_1(x_1)}_{\text{Previous weight}} \underbrace{\frac{\gamma_2(x_{1:2})}{\gamma_1(x_1)q_2(x_2|x_1)}}_{\text{Incremental weight}} \end{aligned}$$

- In particular when the target is the smoothing distribution, we have (due to its recursion) :

$$w_2(x_{1:2}) = w_1(x_1)\frac{f_2(x_2|x_1)g_2(y_2|x_2)}{q_2(x_2|x_1)}$$

Sequential Importance Sampling

Sequential Importance Sampling Algorithm

1: At time 1 : for $j = 1, \ldots, N_p$, sample $X_1^j \sim q_1(x_1)$ and set $w_1^j = \frac{\mu(X_1^j) g_1(y_1|X_1^j)}{q_1(X_1^j)}$

2: **for** time $k > 1$ **do**

3: **for** $j = 1, \ldots, N_p$ **do**

4: Sample $X_k^j \sim q_k(\cdot|X_{k-1}^j)$

5: Compute Importance weight $w_k^j = w_{k-1}^j \frac{g_k(y_k|X_k^j) f_k(X_k^j|X_{k-1}^j)}{q_k(X_k^j|X_{k-1}^j)}$

6: **end for**

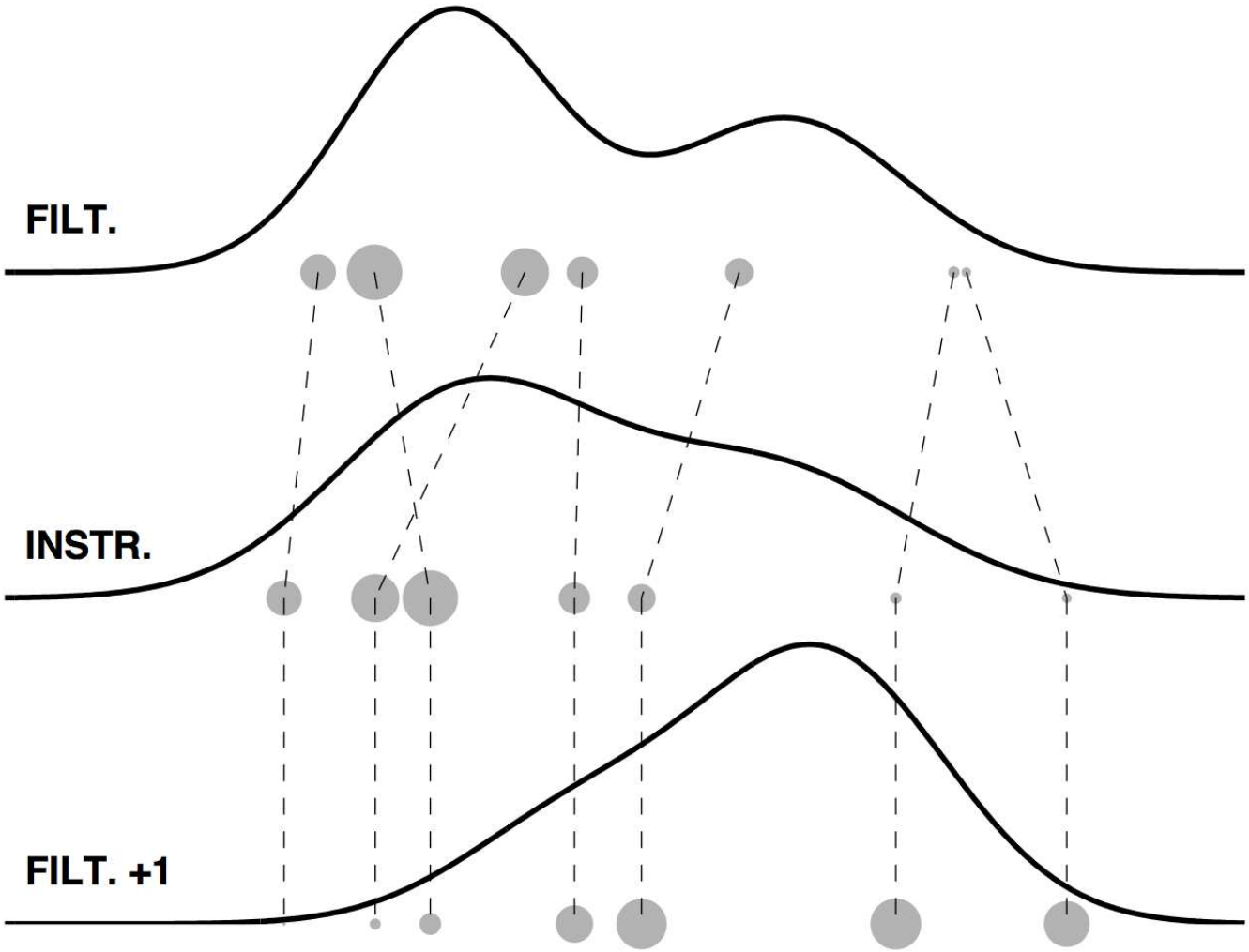
7: Output Approximations :

Filtering : $\pi_k(x_{0:k}) \approx \sum_{i=1}^{N_p} \frac{w_k^i}{\sum_{j=1}^{N_p} w_k^j} \delta_{X_{0:k}^i}(dx_{0:k})$

Normalizing constant : $\widehat{Z}_k = \sum_{j=1}^{N_p} w_k^i$

8: **end for**

Sequential Importance Sampling



One step of the SIS algorithm with just seven particles.

SIS : Choice of the proposal distribution

The so-called "optimal" choice of $q_k(\cdot)$ that minimizes the variance of the importance weights, consists in setting

$$q_k(x_k|x_{k-1}) = \frac{g_k(y_k|x_k)f_k(x_k|x_{k-1})}{\int g_k(y_k|x_k)f_k(x_k|x_{k-1})dx_k}$$

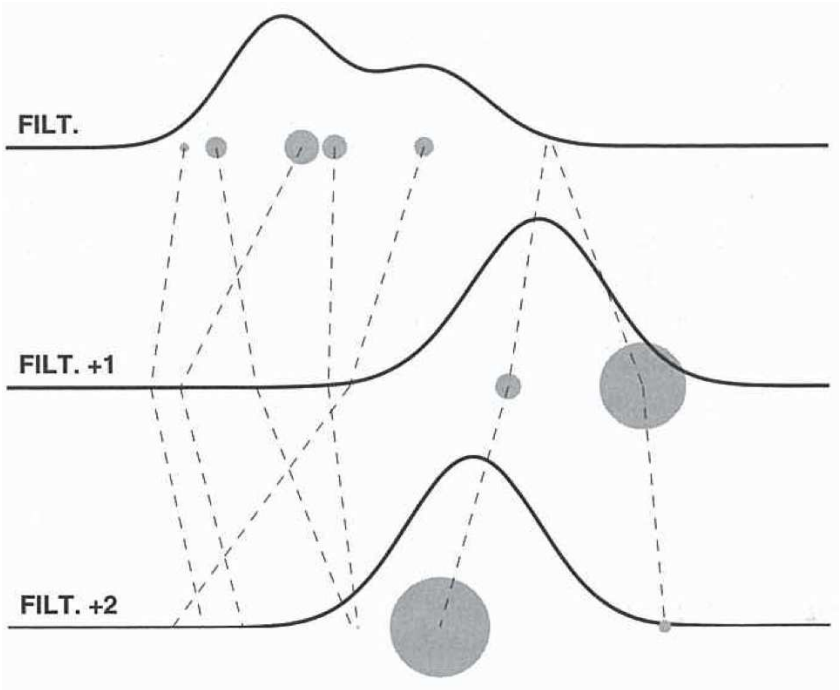
Consequently the weights does not depend on current state value but only on the previous one ($\mathbf{x}_{k-1}$), i.e. :

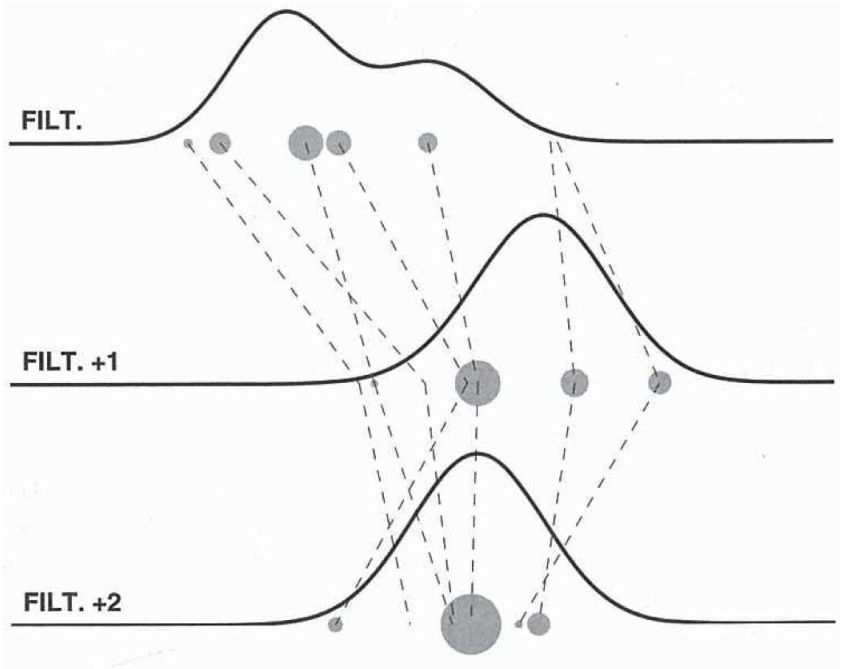
$$w_k^j = w_{k-1}^j \int g_k(y_k|x_k)f_k(x_k|x_{k-1}^j)dx_k$$

This is however usually not feasible and common choices include :

- the prior $q_k(x_k|x_{k-1}) = f_k(x_k|x_{k-1})$ (and then $w_k^j = w_{k-1}^j g_k(y_k|X_k^j)$),
- approximations (sometimes heuristic) to the optimal one (moment matching, use of EKF or UKF, ...),
- tuning parameters of $q_k(\cdot)$ so as to maximize some criterions : effective sample size, entropy, ...

SIS : Choice of the proposal distribution

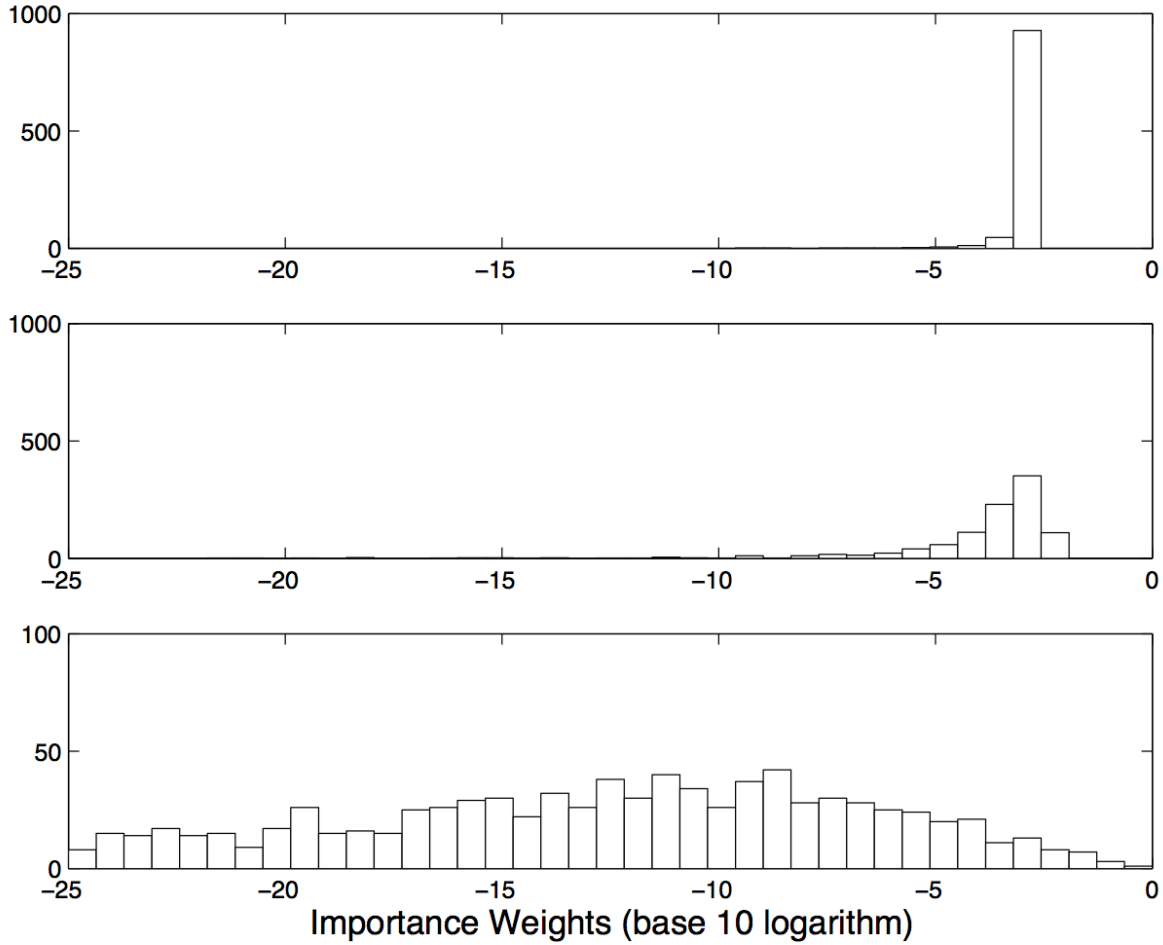




SIS with the of the prior (left) and the optimal (right) distribution as proposal.

$\Rightarrow$ Choice of this proposal distribution is an important step when one want to design an efficient SIS algorithm.

SIS : Weight Degeneracy



Histograms of the base 10 logarithm of the normalized weights for $t = 1$ (top), $t = 50$ (middle) and $t = 100$ (bottom) for a simple stochastic volatility model.

The algorithm performance collapse as t increases... After a few time steps, only a very small number of particles have non negligible weigths ! !

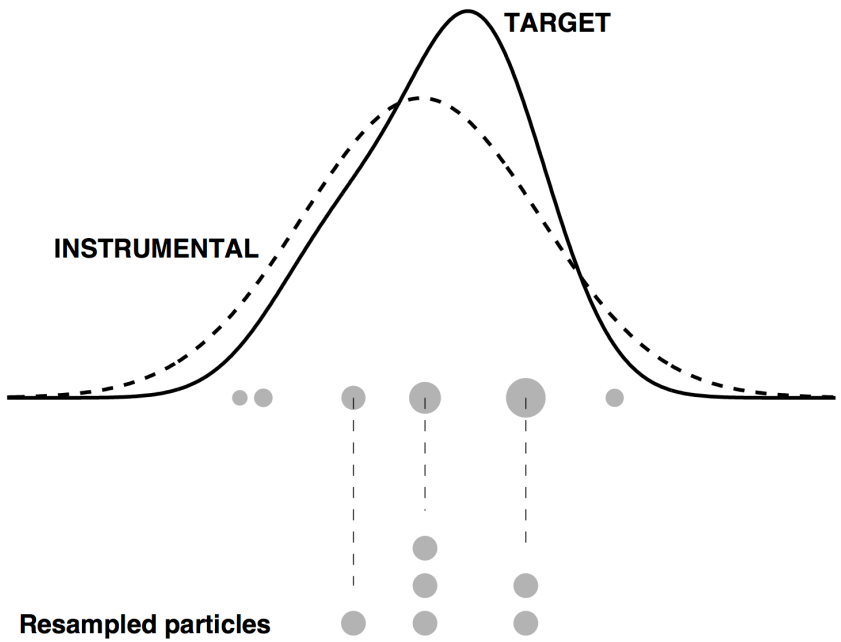
SIS : Resampling Step

Problem : After a few time steps, only a very small number of particles have non negligible weigths !

Solution : Replicate particles with large weights and eliminate those with small weights to prevent the problems we saw with SIS (at the price of a, usually moderate, increase in variance).

$\hookrightarrow$ Use random resampling techniques by taking importance weights : multinomial, residual, ...

$\Rightarrow$ Sequential Importance Resampling

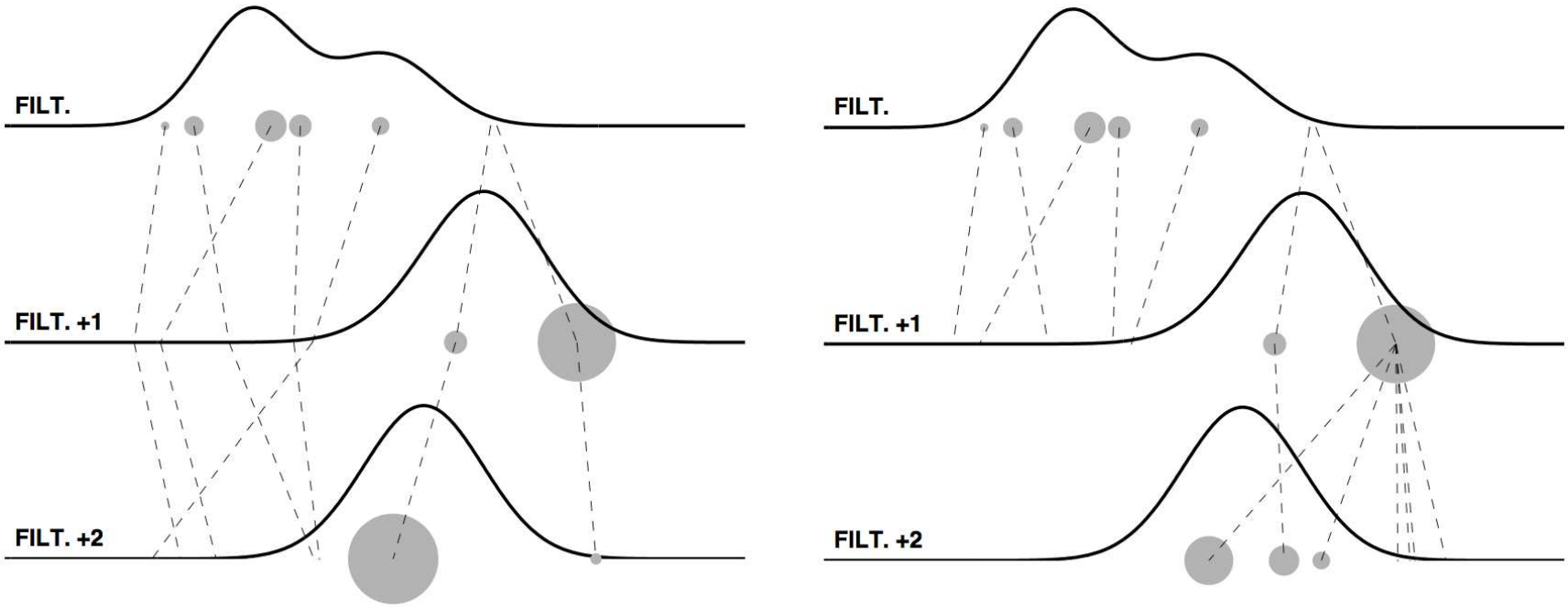


Sequential Importance Sampling Resampling Algorithm

1: At time 1 : for $j = 1, \ldots, N_p$, sample $X_1^j \sim q_1(x_1)$ and set $w_1^j = \frac{\mu(X_1^j) g_1(y_1|X_1^j)}{q_1(X_1^j)}$
2: **for** time $k > 1$ **do**
3: **for** $j = 1, \ldots, N_p$ **do**
4: Sample $X_k^j \sim q_k(\cdot|X_{k-1}^j)$
5: Compute Importance weight $w_k^j = \frac{g_k(y_k|X_k^j) f_k(X_k^j|X_{k-1}^j)}{q_k(X_k^j|X_{k-1}^j)}$
6: **end for**
7: Resampling (if $N_{ess} < \eta$ or every iteration)
8: **end for**

Note : Also known as particle filter

SIS (left) versus SISR (right)

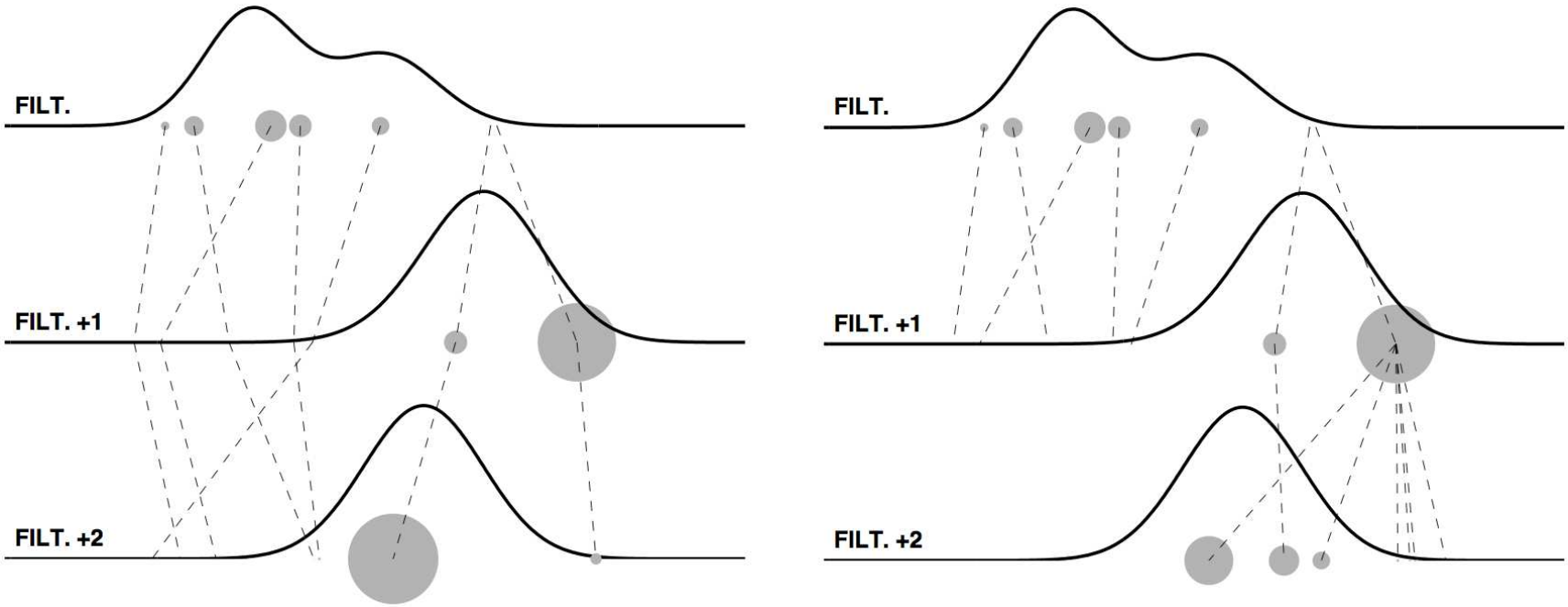


The resampling step is necessary to ensure the long-term stability of the filtering algorithm.

$\hookrightarrow$ Maintain a reasonable number of contributing particles at all times.

BUT it reduces the number of distinct samples $\Rightarrow$ *sample impoverishment*

SIS (left) versus SISR (right)



The resampling step is necessary to ensure the long-term stability of the filtering algorithm.

$\hookrightarrow$ Maintain a reasonable number of contributing particles at all times.

BUT it reduces the number of distinct samples $\Rightarrow$ *sample impoverishment*

SMC for high-dimensional problems

What happens when SMC is applied for high-dimensional problems ?

Let us consider the following filtering problem, when at time k

Prior distribution: $f_k(x_k|x_{k-1})$ Likelihood distribution: $g_k(y_k|x_k)$

where $x_k \in \mathbb{R}^d$ and $y_k \in \mathbb{R}^{d_y}$ with **large values** for d (and d_y).

All SMC variants used the following two steps :

1. Sample $x_k = [x_{k,1}, \ldots, x_{k,n_\mathbf{x}}] \sim q(\cdot|x_{k-1})$
2. Computation of the weights : $w_k = w_{k-1} \frac{g_k(y_k|x_k) f_k(x_k|x_{k-1})}{q_k(x_k|x_{k-1})}$

Limitations of SMC in high-dimensional problems :

- Difficult to design an efficient importance distribution when the space of x_k is high
- High variance in the importance weights (few particles will have non zero weights) due to the evaluation of high-dimensional likelihood

$\Rightarrow$ *"weight degeneracy"* as d (and d_y) increases

[Snyder et al., 2008, Rebeschini and van Handel, 2015]

SMC for high-dimensional problems

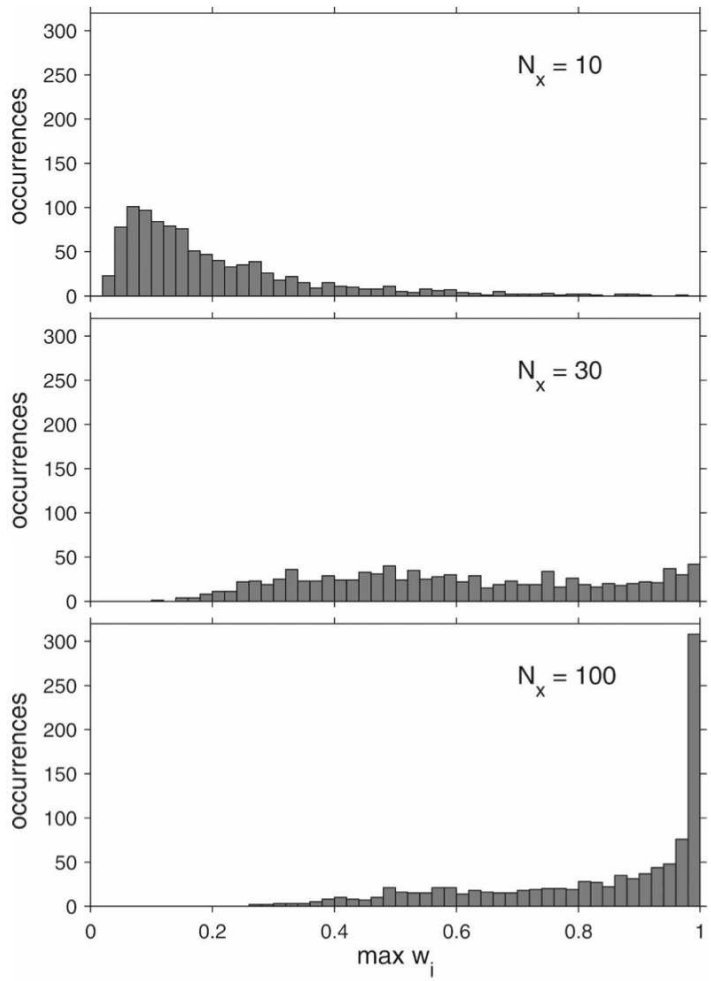
What happens for SMC in high-dimensional problems ?

Illustration with linear and Gaussian model ($d = d_y$) from [Snyder et al., 2008]

$$\begin{cases} \text{Prior :} & x \sim \mathcal{N}(0, \mathbf{I}_d) \\ \text{Likelihood :} & y|x \sim \mathcal{N}(x, \mathbf{I}_d) \end{cases}$$

$\Rightarrow$ Application of one time iter. of the Particle filter ($\Leftrightarrow$ Importance Sampler)

Histograms of $\max \tilde{w}^i$ *for* $d = 10$ *, 30 and 100 with* $N_p = 10^3$ *particles (*10^3 *MC simulations).*



$\Rightarrow$ as $d = d_y$ inscreases, only a very small number of particles have non negligible weigths !

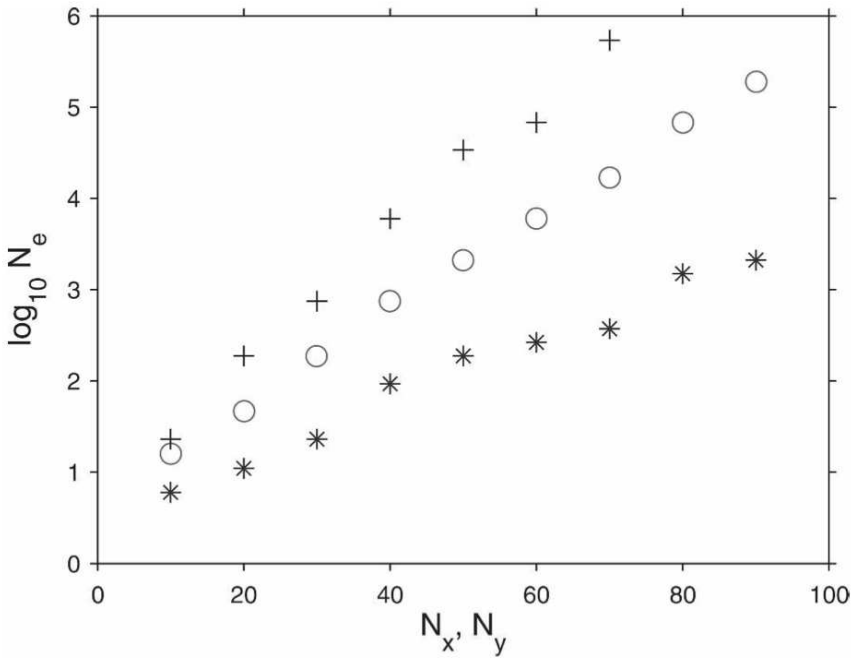
SMC for high-dimensional problems

What happens for SMC in high-dimensional problems ?

Illustration with linear and Gaussian model ($d = d_y$) from [Snyder et al., 2008]

$$\begin{cases} \text{Prior :} & x \sim \mathcal{N}(0, \mathbf{I}_d) \\ \text{Likelihood :} & y|x \sim \mathcal{N}(x, \mathbf{I}_d) \end{cases}$$

$\Rightarrow$ Application of one time iter. of the Particle filter ($\Leftrightarrow$ Importance Sampler)



Number of particles ($\log_{10} N_p$) as a function of d (or d_y) such that $\max \tilde{w}^i$ averaged over 400 realizations is less than 0.6 ($+$), 0.7 ($\circ$), and 0.8 ($*$).

$\Rightarrow$ The required number of particles N_p increases approximately exponentially with d (or d_y) !

SMC for high-dimensional problems

Few recent strategies have been proposed to overcome the curse of dimensionality :

- Block Particle Filter : [Rebeschini and van Handel, 2015]
- Space-time Particle Filter : [Beskos et al., 2014]

but none of these approaches really solve all of the challenges discussed above

Are there any efficient alternatives to SMC for sequential Bayesian inference in High Dimensional Spaces ?

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

Outline

Outline

Sequential MCMC : Introduction

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

- $\rightsquigarrow$ more effective in high-dimensional systems,
- $\rightsquigarrow$ 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]

Contributions of this work :

- Provide a unifying framework for this Sequential MCMC methods,
- Discuss the choice of MCMC kernel,
- Propose efficient strategies based on either Langevin diffusion or Hamiltonian dynamics for filtering.

Sequential MCMC : Introduction

Why MCMC methods are more effective in high dimensional 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 review some 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 $X^n = X^{n-1}$

MCMC : Illustration Metropolis-Hastings

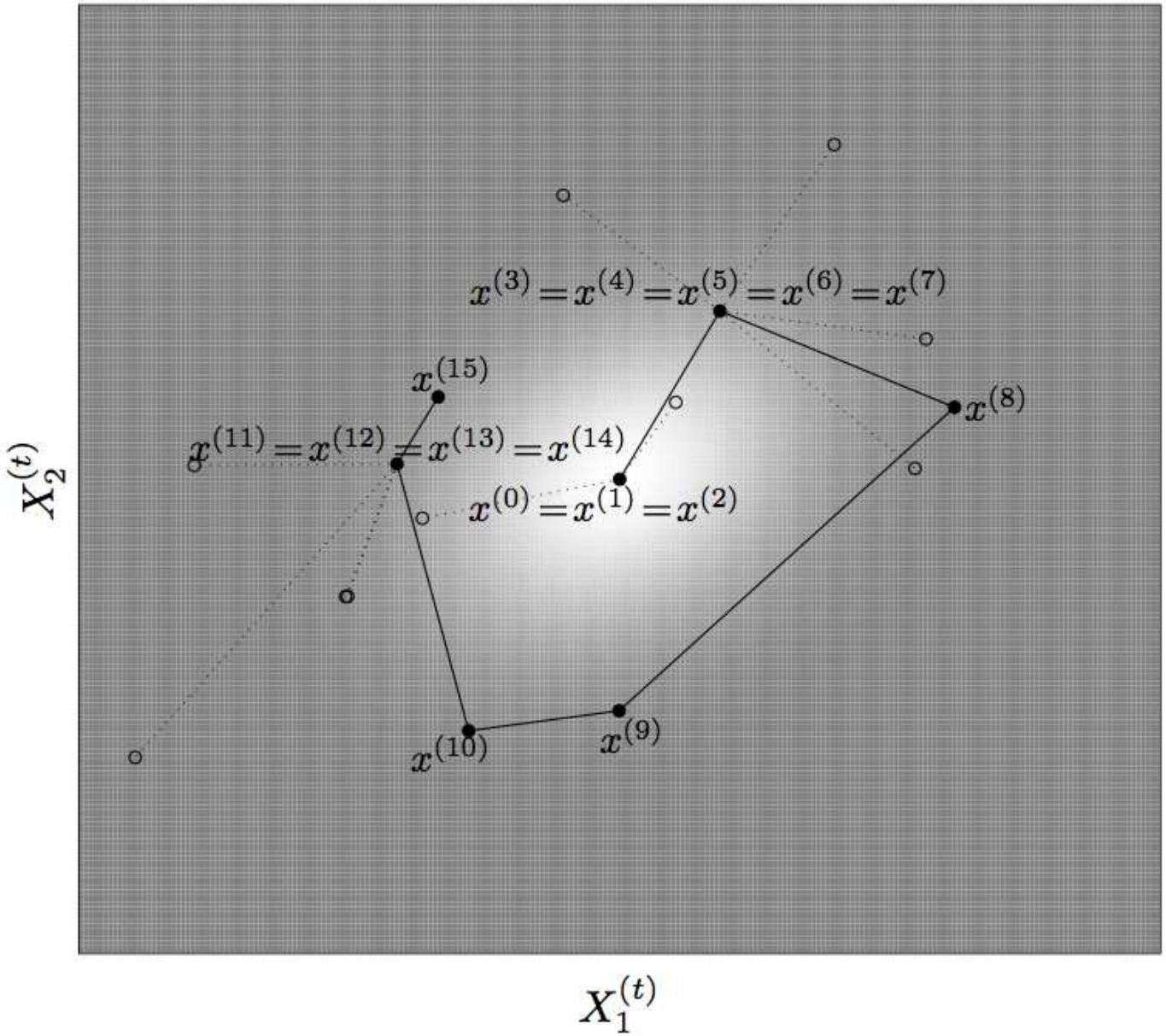


Illustration with a two-dimensional state ($d = 2$)

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 $\gamma(X^*)/\gamma(X^{n-1}) < 1$

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.

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

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 $\pi_{n-1}(x_{1:n-1})$.

$\Rightarrow$ **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 $\pi_n(x_{1:n})$ as defined in Eq. (2).**

Sequential MCMC : General Principle

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

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

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.

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

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 $\pi_{n-1}(x_{1:n-1})$.

$\Rightarrow$ 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 $\pi_n(x_{1:n})$ as defined in Eq. (2).

Sequential MCMC : General Principle

General SMCMC for filtering

1. <u>If time $n = 1$</u>
2. For $j = 1, \ldots, N + N_b$
3. Sample $X_{1,1}^j \sim \mathcal{K}_1(X_{1,1}^{j-1}, \cdot)$ 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. <u>Elseif time $n \geq 2$</u>
5. For $j = 1, \ldots, N + N_b$
6. *[OPTIONAL]* Refine empirical approximation of previous posterior distributions as described in [Brockwell et al., 2010]
7. Sample $X_{n,1:n}^j \sim \mathcal{K}_n(X_{n,1:n}^{j-1}, \cdot)$ with $\mathcal{K}_n$ an MCMC kernel of invariant distribution π_n defined in Eq. (2).
8. **Output :** Approximation of the smoothing distribution with the following empirical measure :

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

Optimal Independent Metropolis-Hastings Kernel

This choice consists in using the following proposal distribution :

$$\begin{aligned} q(x_{1:n}|X_{1:n}^{i-1}) &= q(x_{1:n}) = \pi_n(x_{1:n}), \\ &\propto g_n(y_n|x_n) f_n(x_n|x_{n-1}) \sum_{m=N_b+1}^{N_b+N} \delta_{X_{n-1,1:n-1}^m}(dx_{1:n-1}), \\ &\propto p(x_n|y_n, x_{n-1}) \sum_{m=N_b+1}^{N_b+N} p(y_n|x_{n-1} = X_{n-1,n-1}^m) \delta_{X_{n-1,1:n-1}^m}(dx_{1:n-1}). \end{aligned} \tag{4}$$

from which a sample can be obtained by following these two steps :

1. Generate $X_{n,1:n-1}^* \sim \sum_{m=N_b+1}^{N_b+N} \alpha^m \delta_{X_{n-1,1:n-1}^m}(dx_{1:n-1})$ with
$$\alpha^m = \frac{p(y_n|x_{n-1} = X_{n-1,n-1}^m)}{\sum_{j=N_b+1}^{N_b+N} p(y_n|x_{n-1} = X_{n-1,n-1}^j)}$$
2. Generate $X_{n,n}^* \sim p(x_n|y_n, X_{n,n-1}^*)$

Unfortunately (as for SMC), impossible in most scenarios both :

- to sample from $p(x_n|y_n, x_{n-1})$
- to evaluate $p(y_n|x_{n-1}) = \int_{\mathbb{R}^d} g_n(y_n|x_n) f_n(x_n|x_{n-1}) dx_n$.

Optimal Independent Metropolis-Hastings Kernel

This choice consists in using the following proposal distribution :

$$\begin{aligned} q(x_{1:n}|X_{1:n}^{i-1}) &= q(x_{1:n}) = \pi_n(x_{1:n}), \\ &\propto g_n(y_n|x_n) f_n(x_n|x_{n-1}) \sum_{m=N_b+1}^{N_b+N} \delta_{X_{n-1,1:n-1}^m}(dx_{1:n-1}), \\ &\propto p(x_n|y_n, x_{n-1}) \sum_{m=N_b+1}^{N_b+N} p(y_n|x_{n-1} = X_{n-1,n-1}^m) \delta_{X_{n-1,1:n-1}^m}(dx_{1:n-1}). \end{aligned} \tag{4}$$

from which a sample can be obtained by following these two steps :

1. Generate $X_{n,1:n-1}^* \sim \sum_{m=N_b+1}^{N_b+N} \alpha^m \delta_{X_{n-1,1:n-1}^m}(dx_{1:n-1})$ with
$$\alpha^m = \frac{p(y_n|x_{n-1} = X_{n-1,n-1}^m)}{\sum_{j=N_b+1}^{N_b+N} p(y_n|x_{n-1} = X_{n-1,n-1}^j)}$$
2. Generate $X_{n,n}^* \sim p(x_n|y_n, X_{n,n-1}^*)$

Unfortunately (as for SMC), impossible in most scenarios both :

- to sample from $p(x_n|y_n, x_{n-1})$
- to evaluate $p(y_n|x_{n-1}) = \int_{\mathbb{R}^d} g_n(y_n|x_n) f_n(x_n|x_{n-1}) dx_n$.

Alternative Choices for Independent MH

- Approximate the optimal one discussed previously (Normal approx, linearization, ...)
- Use a combination of the prior distribution and the empirical approximation of the previous posterior :

$$f_n(x_n|x_{n-1})\frac{1}{N}\sum_{m=N_b+1}^{N_b+N}\delta_{X^m_{n-1,1:n-1}}(dx_{1:n-1})$$

However, this could be very inefficient, especially in high-dimensional systems

- Block sampling using a series of Metropolis-within Gibbs schemes

Problem : poor mixing rate in the presence of strong correlation

Proposition :

- Use the ability of MCMC to design local moves : $X^*_{1:n} \sim q(x_{1:n}|X^{i-1}_{1:n})$
- Use gradient information from the target within the MCMC kernel to traverse the space efficiently

Alternative Choices for Independent MH

- Approximate the optimal one discussed previously (Normal approx, linearization, ...)
- Use a combination of the prior distribution and the empirical approximation of the previous posterior :

$$f_n(x_n|x_{n-1})\frac{1}{N}\sum_{m=N_b+1}^{N_b+N}\delta_{X^m_{n-1,1:n-1}}(dx_{1:n-1})$$

However, this could be very inefficient, especially in high-dimensional systems

- Block sampling using a series of Metropolis-within Gibbs schemes

Problem : poor mixing rate in the presence of strong correlation

Proposition :

- Use the ability of MCMC to design local moves : $X^*_{1:n} \sim q(x_{1:n}|X^{i-1}_{1:n})$
- Use gradient information from the target within the MCMC kernel to traverse the space efficiently

Alternative Choices for Independent MH

- Approximate the optimal one discussed previously (Normal approx, linearization, ...)
- Use a combination of the prior distribution and the empirical approximation of the previous posterior :

$$f_n(x_n|x_{n-1})\frac{1}{N}\sum_{m=N_b+1}^{N_b+N}\delta_{X^m_{n-1,1:n-1}}(dx_{1:n-1})$$

However, this could be very inefficient, especially in high-dimensional systems

- Block sampling using a series of Metropolis-within Gibbs schemes

Problem : poor mixing rate in the presence of strong correlation

Proposition :

- Use the ability of MCMC to design local moves : $X^*_{1:n} \sim q(x_{1:n}|X^{i-1}_{1:n})$
- Use gradient information from the target within the MCMC kernel to traverse the space efficiently

Alternative Choices for Independent MH

- Approximate the optimal one discussed previously (Normal approx, linearization, ...)
- Use a combination of the prior distribution and the empirical approximation of the previous posterior :

$$f_n(x_n|x_{n-1})\frac{1}{N}\sum_{m=N_b+1}^{N_b+N}\delta_{X^m_{n-1,1:n-1}}(dx_{1:n-1})$$

However, this could be very inefficient, especially in high-dimensional systems

- Block sampling using a series of Metropolis-within Gibbs schemes

Problem : poor mixing rate in the presence of strong correlation

Proposition :

- Use the ability of MCMC to design local moves : $X^*_{1:n} \sim q(x_{1:n}|X^{i-1}_{1:n})$
- Use gradient information from the target within the MCMC kernel to traverse the space efficiently

Alternative Choices for Independent MH

- Approximate the optimal one discussed previously (Normal approx, linearization, ...)
- Use a combination of the prior distribution and the empirical approximation of the previous posterior :

$$f_n(x_n|x_{n-1})\frac{1}{N}\sum_{m=N_b+1}^{N_b+N}\delta_{X^m_{n-1,1:n-1}}(dx_{1:n-1})$$

However, this could be very inefficient, especially in high-dimensional systems

- Block sampling using a series of Metropolis-within Gibbs schemes

Problem : poor mixing rate in the presence of strong correlation

Proposition :

- Use the ability of MCMC to design local moves : $X^*_{1:n} \sim q(x_{1:n}|X^{i-1}_{1:n})$
- Use gradient information from the target within the MCMC kernel to traverse the space efficiently

Langevin and Hamiltonian based SMCMC

Use MCMC kernel families based on Langevin diffusion and Hamiltonian dynamics
↪ Use gradient information in a different way to traverse a **continuous** space efficiently.
However, the target distribution is

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

- ⇝ Discrete space for $x_{1:n-1}$
- ⇝ Continuous space for x_n

Proposition : at time step n and the j-th iteration of the MCMC, a succession of the two MH-within Gibbs steps :

1) Sample $X^j_{n,1:n-1}$ from the set $\{X^m_{n-1,1:n-1}\}_{m=N_b+1}^{N_b+N}$ given $X^{j-1}_{n,1:n}$

Langevin and Hamiltonian based SMCMC

Use MCMC kernel families based on Langevin diffusion and Hamiltonian dynamics
↪ Use gradient information in a different way to traverse a **continuous** space efficiently.
However, the target distribution is

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

⇝ Discrete space for $x_{1:n-1}$

⇝ Continuous space for x_n

Proposition : at time step n and the j-th iteration of the MCMC, a succession of the two MH-within Gibbs steps :

1) Sample $X^j_{n,1:n-1}$ from the set $\{X^m_{n-1,1:n-1}\}_{m=N_b+1}^{N_b+N}$ given $X^{j-1}_{n,1:n}$ using either

- an uniform draw and then accept with proba $\alpha = f_n(X^{j-1}_{n,n}|X^*_{n,n-1}) / f_n(X^{j-1}_{n,n}|X^{j-1}_{n,n-1})$
- draw from

$$X^j_{n,1:n-1} \sim \sum_{m=N_b+1}^{N_b+N} \frac{f_n(x_n = X^{j-1}_{n,n}|x_{n-1} = X^m_{n-1,n-1})}{\sum_i f_n(x_n = X^{j-1}_{n,n}|x_{n-1} = X^i_{n-1,n-1})} \delta_{X^m_{n-1,1:n-1}}(dx_{1:n-1})$$

Langevin and Hamiltonian based SMCMC

Use MCMC kernel families based on Langevin diffusion and Hamiltonian dynamics
↪ Use gradient information in a different way to traverse a **continuous** space efficiently.
However, the target distribution is

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

- ⇝ Discrete space for $x_{1:n-1}$
- ⇝ Continuous space for x_n

Proposition : at time step n and the j-th iteration of the MCMC, a succession of the two MH-within Gibbs steps :

1) Sample $X^j_{n,1:n-1}$ from the set $\{X^m_{n-1,1:n-1}\}_{m=N_b+1}^{N_b+N}$ given $X^{j-1}_{n,1:n}$
2) Sample $X^j_{n,n}$ given $X^{j-1}_{n,n}$ and $X^j_{n,1:n-1}$ using either Langevin diffusion or Hamiltonian dynamics based MH kernel.

The target distribution of the second step is thus given by the following conditional posterior on a continuous space :

$$\widetilde{\pi}(x_n) = \pi_n(x_n|X^j_{n,1:n-1}) \propto g_n(y_n|x_n) f_n(x_n|x_{n-1} = X^j_{n,n-1}). \tag{5}$$

Langevin diffusion based MCMC kernel

Langevin diffusion is given by the solution of the following stochastic differential equation (SDE)

$$dX^t = \frac{1}{2}\nabla \log \widetilde{\pi}(X^t)dt + dB^t \text{ with } B^t \text{ a standard Brownian motion} \quad (6)$$

It represents a process with stationary and limiting distribution $\widetilde{\pi}$

A direct use of this SDE by using a first-order Euler discretization as in [Ermak, 1975] gives a proposal mechanism that creates the following Markov chain

$$X^{i+1}|X^i \sim q(x|X^i) = \mathcal{N}\left(x \middle| X^i + \frac{\epsilon^2}{2}\nabla \log \widetilde{\pi}(X^i)\ , \epsilon^2 \mathbf{I}_d\right) \quad (7)$$

with ϵ the integration step size.
Unfortunately, convergence of the Markov chain is no longer guaranteed for finite step size ϵ

$\Rightarrow$ [Rossky et al., 1978] proposes a Metropolized version which ensures convergence to the invariant measure.
$\Rightarrow$ Metropolis Adjusted Langevin Algorithm (MALA) :

1. Sample $X^* \sim q(x|X^i)$ as defined in Eq. (7)
2. Acceptance step with probability $\min\left(1, \widetilde{\pi}(X^*)q(X^i|X^*)/\widetilde{\pi}(X^i)q(X^*|X^i)\right)$

Langevin diffusion based MCMC kernel

Langevin diffusion is given by the solution of the following stochastic differential equation (SDE)

$$dX^t = \frac{1}{2}\nabla \log \widetilde{\pi}(X^t)dt + dB^t \text{ with } B^t \text{ a standard Brownian motion} \quad (6)$$

It represents a process with stationary and limiting distribution $\widetilde{\pi}$

A direct use of this SDE by using a first-order Euler discretization as in [Ermak, 1975] gives a proposal mechanism that creates the following Markov chain

$$X^{i+1}|X^i \sim q(x|X^i) = \mathcal{N}\left(x \middle| X^i + \frac{\epsilon^2}{2}\nabla \log \widetilde{\pi}(X^i)\ , \epsilon^2 \mathbf{I}_d\right) \quad (7)$$

with ϵ the integration step size.
Unfortunately, convergence of the Markov chain is no longer guaranteed for finite step size ϵ

$\Rightarrow$ [Rossky et al., 1978] proposes a Metropolized version which ensures convergence to the invariant measure.
$\Rightarrow$ Metropolis Adjusted Langevin Algorithm (MALA) :

1. Sample $X^* \sim q(x|X^i)$ as defined in Eq. (7)
2. Acceptance step with probability $\min\left(1, \widetilde{\pi}(X^*)q(X^i|X^*)/\widetilde{\pi}(X^i)q(X^*|X^i)\right)$

Langevin diffusion based MCMC kernel

Improvement of MALA

- Use a constant pre-defined covariance Σ :

$$q(x|X^i) = \mathcal{N}\left(x \middle| X^i + \frac{\epsilon^2}{2}\Sigma\nabla \log \widetilde{\pi}(X^i)\ , \epsilon^2\Sigma\right), \tag{8}$$

  leading to the "pre-conditioned" MALA [Roberts and Stramer, 2002].

- Langevin diffusion on a Riemannian manifold [Girolami and Calderhead, 2011, Xifara et al., 2014, Livingstone and Girolami, 2014].
  $\rightsquigarrow$ Take into account the local structure of the target density when proposing (speed up the convergence of the Markov chain).
  $\Rightarrow$ Consists in adopting a position specific covariance.

$$q(x|X^i) = \mathcal{N}\left(x \middle| X^i + \frac{\epsilon^2}{2}G^{-1}(X^i)\nabla \log \widetilde{\pi}(X^i) + \frac{\epsilon^2}{2}\Lambda(X^i), \epsilon^2 G^{-1}(X^i)\right),$$

$$\text{with } \Lambda_i(X^t) = \sum_{j=1}^{d} \frac{\partial}{\partial x(j)} \left[G^{-1}(X^t)\right]_{ij}$$

- Simplified manifold MALA algorithm (no drift) in which the proposal is given by :

$$q(x|X^i) = \mathcal{N}\left(x \middle| X^i + \frac{\epsilon^2}{2}G^{-1}(X^i)\nabla \log \widetilde{\pi}(X^i)\ , \epsilon^2 G^{-1}(X^i)\right), \tag{9}$$

Langevin diffusion based MCMC kernel

Improvement of MALA

- Use a constant pre-defined covariance Σ :

$$q(x|X^i) = \mathcal{N}\left(x \middle| X^i + \frac{\epsilon^2}{2}\Sigma\nabla \log \widetilde{\pi}(X^i)\ , \epsilon^2\Sigma\right), \tag{8}$$

  leading to the "pre-conditioned" MALA [Roberts and Stramer, 2002].

- Langevin diffusion on a Riemannian manifold [Girolami and Calderhead, 2011, Xifara et al., 2014, Livingstone and Girolami, 2014].
  $\rightsquigarrow$ Take into account the local structure of the target density when proposing (speed up the convergence of the Markov chain).
  $\Rightarrow$ Consists in adopting a position specific covariance.

$$q(x|X^i) = \mathcal{N}\left(x \middle| X^i + \frac{\epsilon^2}{2}G^{-1}(X^i)\nabla \log \widetilde{\pi}(X^i) + \frac{\epsilon^2}{2}\Lambda(X^i), \epsilon^2 G^{-1}(X^i)\right),$$

$$\text{with } \Lambda_i(X^t) = \sum_{j=1}^{d} \frac{\partial}{\partial x(j)}\left[G^{-1}(X^t)\right]_{ij}$$

- Simplified manifold MALA algorithm (no drift) in which the proposal is given by :

$$q(x|X^i) = \mathcal{N}\left(x \middle| X^i + \frac{\epsilon^2}{2}G^{-1}(X^i)\nabla \log \widetilde{\pi}(X^i)\ , \epsilon^2 G^{-1}(X^i)\right), \tag{9}$$

Improvement of MALA

- Use a constant pre-defined covariance Σ :

$$q(x|X^i) = \mathcal{N}\left(x \middle| X^i + \frac{\epsilon^2}{2}\Sigma\nabla \log \widetilde{\pi}(X^i)\ , \epsilon^2\Sigma\right), \tag{8}$$

  leading to the "pre-conditioned" MALA [Roberts and Stramer, 2002].

- Langevin diffusion on a Riemannian manifold [Girolami and Calderhead, 2011, Xifara et al., 2014, Livingstone and Girolami, 2014].
  $\rightsquigarrow$ Take into account the local structure of the target density when proposing (speed up the convergence of the Markov chain).
  $\Rightarrow$ Consists in adopting a position specific covariance.

$$q(x|X^i) = \mathcal{N}\left(x \middle| X^i + \frac{\epsilon^2}{2}G^{-1}(X^i)\nabla \log \widetilde{\pi}(X^i) + \frac{\epsilon^2}{2}\Lambda(X^i), \epsilon^2 G^{-1}(X^i)\right),$$

$$\text{with } \Lambda_i(X^t) = \sum_{j=1}^{d} \frac{\partial}{\partial x(j)}\left[G^{-1}(X^t)\right]_{ij}$$

- Simplified manifold MALA algorithm (no drift) in which the proposal is given by :

$$q(x|X^i) = \mathcal{N}\left(x \middle| X^i + \frac{\epsilon^2}{2}G^{-1}(X^i)\nabla \log \widetilde{\pi}(X^i)\ , \epsilon^2 G^{-1}(X^i)\right), \tag{9}$$

Hamiltonian diffusion based MCMC kernel

- Hamiltonian dynamics was originally introduced in molecular simulation and later was used within an MCMC framework in [Duane et al., 1987] leading to the so-called "Hybrid Monte Carlo"
- HMC is a powerful methodology to sample from a continuous distribution by introducing an auxiliary variable, $q \in \mathbb{R}^d$ called *momentum variables*
- The Hamiltonian function in our case is defined by

$$H(x,q) = \underbrace{-\log \widetilde{\pi}(x)}_{U(x)} + \underbrace{\frac{1}{2} q^T M^{-1} q}_{F(q)}, \tag{10}$$

- The dynamics of both variables with respect to a fictitious time τ are given by the Hamiltonian equations :

$$\frac{\partial x(i)}{\partial \tau} = \frac{\partial H}{\partial q(i)} = \left[M^{-1} q\right]_i \text{ and } \frac{\partial q(i)}{\partial \tau} = -\frac{\partial H}{\partial x(i)} = -\frac{\partial U}{\partial x(i)} = \frac{\partial \log \widetilde{\pi}(x)}{\partial x(i)}.$$

- Hamiltonian dynamics posses some interesting properties (energy and volume preservation as well as time reversibility - see [Neal, 2010]), that allow its use in constructing MCMC kernel.
- The Hamiltonian in Eq. (10) defines equivalently the following joint distribution :

$$\widetilde{\pi}(x,q) \propto \exp\left(-H(x,q)\right) = \widetilde{\pi}(x) \exp\left(-\frac{1}{2} q^T M^{-1} q\right), \tag{11}$$

which admits as marginal the target distribution of interest $\widetilde{\pi}(x)$.

Hamiltonian diffusion based MCMC kernel

- Hamiltonian dynamics was originally introduced in molecular simulation and later was used within an MCMC framework in [Duane et al., 1987] leading to the so-called "Hybrid Monte Carlo"
- HMC is a powerful methodology to sample from a continuous distribution by introducing an auxiliary variable, $q \in \mathbb{R}^d$ called *momentum variables*
- The Hamiltonian function in our case is defined by

$$H(x,q) = \underbrace{-\log \widetilde{\pi}(x)}_{U(x)} + \underbrace{\frac{1}{2} q^T M^{-1} q}_{F(q)}, \tag{10}$$

- The dynamics of both variables with respect to a fictitious time τ are given by the Hamiltonian equations :

$$\frac{\partial x(i)}{\partial \tau} = \frac{\partial H}{\partial q(i)} = \left[M^{-1} q\right]_i \text{ and } \frac{\partial q(i)}{\partial \tau} = -\frac{\partial H}{\partial x(i)} = -\frac{\partial U}{\partial x(i)} = \frac{\partial \log \widetilde{\pi}(x)}{\partial x(i)}.$$

- Hamiltonian dynamics posses some interesting properties (energy and volume preservation as well as time reversibility - see [Neal, 2010]), that allow its use in constructing MCMC kernel.
- The Hamiltonian in Eq. (10) defines equivalently the following joint distribution :

$$\widetilde{\pi}(x,q) \propto \exp\left(-H(x,q)\right) = \widetilde{\pi}(x) \exp\left(-\frac{1}{2} q^T M^{-1} q\right), \tag{11}$$

which admits as marginal the target distribution of interest $\widetilde{\pi}(x)$.

Hamiltonian diffusion based MCMC kernel

- Hamiltonian dynamics was originally introduced in molecular simulation and later was used within an MCMC framework in [Duane et al., 1987] leading to the so-called "Hybrid Monte Carlo"
- HMC is a powerful methodology to sample from a continuous distribution by introducing an auxiliary variable, $q \in \mathbb{R}^d$ called *momentum variables*
- The Hamiltonian function in our case is defined by

$$H(x,q) = \underbrace{-\log \widetilde{\pi}(x)}_{U(x)} + \underbrace{\frac{1}{2} q^T M^{-1} q}_{F(q)}, \tag{10}$$

- The dynamics of both variables with respect to a fictitious time τ are given by the Hamiltonian equations :

$$\frac{\partial x(i)}{\partial \tau} = \frac{\partial H}{\partial q(i)} = \left[M^{-1} q\right]_i \text{ and } \frac{\partial q(i)}{\partial \tau} = -\frac{\partial H}{\partial x(i)} = -\frac{\partial U}{\partial x(i)} = \frac{\partial \log \widetilde{\pi}(x)}{\partial x(i)}.$$

- Hamiltonian dynamics posses some interesting properties (energy and volume preservation as well as time reversibility - see [Neal, 2010]), that allow its use in constructing MCMC kernel.
- The Hamiltonian in Eq. (10) defines equivalently the following joint distribution :

$$\widetilde{\pi}(x,q) \propto \exp\left(-H(x,q)\right) = \widetilde{\pi}(x) \exp\left(-\frac{1}{2} q^T M^{-1} q\right), \tag{11}$$

which admits as marginal the target distribution of interest $\widetilde{\pi}(x)$.

Hamiltonian diffusion based MCMC kernel

Hamiltonian based MCMC Kernel for sampling $X^j_{n,n}$ in the SMCMC

1. Sample $Q^j \sim \widetilde{\pi}(q|X^{j-1}) = \mathcal{N}(q\,|\mathbf{0}, M\,)$
2. Propose (X^*, Q^*) using the Leapfrog method with (X^{j-1}, Q^j) as initial values.
3. Compute the MH acceptance probability $\rho_{\mathsf{HMC}} = \min\left\{1, \exp\left(-H(X^*, Q^*) + H(X^{j-1}, Q^j)\right)\right\}$
4. Accept $X^j = X^*$ with probability ρ_{HMC} otherwise set $X^j = X^{j-1}$

- Hamiltonian dynamics are numerically simulated using a discretization method named the Leapfrog method
- Acceptance rule in order to correct the fact that the leapfrog method induces a bias
- To avoid possible periodic trajectories of the HMC thus leading to a non-ergodic algorithm, it is recommended to randomly choose either the step size ϵ or the the number of leapfrog steps N_{LF}[Neal, 2010]

Hamiltonian diffusion based MCMC kernel

Hamiltonian based MCMC Kernel for sampling $X^j_{n,n}$ in the SMCMC

1. Sample $Q^j \sim \widetilde{\pi}(q|X^{j-1}) = \mathcal{N}(q\,|\mathbf{0}, M\,)$
2. Propose (X^*, Q^*) using the Leapfrog method with (X^{j-1}, Q^j) as initial values.
3. Compute the MH acceptance probability $\rho_{\mathsf{HMC}} = \min\left\{1, \exp\left(-H(X^*, Q^*) + H(X^{j-1}, Q^j)\right)\right\}$
4. Accept $X^j = X^*$ with probability ρ_{HMC} otherwise set $X^j = X^{j-1}$

- Using a single step integrator ($N_{LF} = 1$) with the Leapfrog method is exactly equivalent to the pre-conditioned MALA algorithm
  but their properties are quite different.
- MALA is a random-walk MH adjusted by taking into account the gradient-based information whereas the HMC proposal involves a deterministic element based on Hamiltonian equation
  - $\rightsquigarrow$ With an appropriate tuning of its parameters (N_{LF} and ϵ), the HMC is able to reach a state that is almost independent of the current Markov state
- asymptotic analysis of these algorithms : in the stationary regime, the random-walk MH algorithm needs $\mathcal{O}(d)$ steps to explore the state space whereas MALA and HMC needs only $\mathcal{O}(d^{1/3})$ and $\mathcal{O}(d^{1/4})$, respectively [Green et al., 2015]

Improvement of HMC

- [Girolami and Calderhead, 2011] proposes a generalization of this HMC algorithm by considering Hamiltonian dynamics on a manifold
  $\rightsquigarrow$ take into account the local structure of the target distribution.
- The Hamiltonian is now defined as :

$$\begin{aligned} \widetilde{H}(x,q) &= U(x) + \widetilde{F}(q,x), \\ \text{with } U(x) &= -\log \widetilde{\pi}(x) \\ \text{and } \widetilde{F}(q,x) &= \frac{1}{2}\log\left((2\pi)^d |G(x)|\right) + \frac{1}{2} q^T G^{-1}(x) q. \end{aligned} \tag{12}$$

- The Hamiltonian is no longer separable and therefore the Hamiltonian dynamics of each variable will now depend on both variables, i.e.

$$\frac{\partial x(i)}{\partial \tau} = \frac{\partial \widetilde{H}}{\partial q(i)} = \left[G^{-1}(x)q\right]_i \tag{13}$$

and

$$\begin{aligned} \frac{\partial q(i)}{\partial \tau} &= -\frac{\partial \widetilde{H}}{\partial x(i)} = -\frac{\partial U(x)}{\partial x(i)} - \frac{1}{2}\frac{\partial \log(|G(x)|)}{\partial x(i)} - \frac{1}{2} q^T \frac{\partial G^{-1}(x)}{\partial x(i)} q \\ &= \frac{\partial \log \widetilde{\pi}(x)}{\partial x(i)} - \frac{1}{2}\mathrm{Tr}\left\{G^{-1}(x)\frac{\partial G(x)}{\partial x(i)}\right\} + \frac{1}{2} q^T G^{-1}(x)\frac{\partial G(x)}{\partial x(i)} G^{-1}(x) q \end{aligned} \tag{14}$$

$\Rightarrow$ To numerically simulate these Hamiltonian dynamics on a manifold, a **generalized** version of the Leapfrog integrator has to be used.

Improvement of HMC

- [Girolami and Calderhead, 2011] proposes a generalization of this HMC algorithm by considering Hamiltonian dynamics on a manifold
  $\rightsquigarrow$ take into account the local structure of the target distribution.
- The Hamiltonian is now defined as :

$$\begin{aligned} \widetilde{H}(x,q) &= U(x) + \widetilde{F}(q,x), \\ \text{with } U(x) &= -\log \widetilde{\pi}(x) \\ \text{and } \widetilde{F}(q,x) &= \frac{1}{2}\log\Big((2\pi)^d |G(x)|\Big) + \frac{1}{2} q^T G^{-1}(x) q. \end{aligned} \tag{12}$$

- The Hamiltonian is no longer separable and therefore the Hamiltonian dynamics of each variable will now depend on both variables, i.e.

$$\frac{\partial x(i)}{\partial \tau} = \frac{\partial \widetilde{H}}{\partial q(i)} = \left[G^{-1}(x) q\right]_i \tag{13}$$

and

$$\begin{aligned} \frac{\partial q(i)}{\partial \tau} &= -\frac{\partial \widetilde{H}}{\partial x(i)} = -\frac{\partial U(x)}{\partial x(i)} - \frac{1}{2}\frac{\partial \log(|G(x)|)}{\partial x(i)} - \frac{1}{2} q^T \frac{\partial G^{-1}(x)}{\partial x(i)} q \\ &= \frac{\partial \log \widetilde{\pi}(x)}{\partial x(i)} - \frac{1}{2}\mathsf{Tr}\left\{ G^{-1}(x)\frac{\partial G(x)}{\partial x(i)} \right\} + \frac{1}{2} q^T G^{-1}(x) \frac{\partial G(x)}{\partial x(i)} G^{-1}(x) q \end{aligned} \tag{14}$$

$\Rightarrow$ To numerically simulate these Hamiltonian dynamics on a manifold, a **generalized** version of the Leapfrog integrator has to be used.

Choice of the tensor metric $G(\cdot)$

- A natural choice for this metric is to take into account the local structure of the target distribution by using information from its hessian, i.e.

$$G(x_n) = -\Delta_{x_n}^{x_n} \log \widetilde{\pi}(x_n), \tag{15}$$

where $\Delta_{x_n}^{x_n} := \nabla_{x_n} \nabla_{x_n}^T$ is the second derivative operator.
$\Rightarrow$ Use of local curvature of the target distribution

- **one major issue** with this choice (unless the target distribution is log-concave) this negative Hessian will not be globally positive-definite
- [Betancourt, 2013] proposes a smooth absolute transformation of the eigenvalues that maps this negative Hessian metric into a positive-definite matrix in a way that the derivative of this transformed metric (required in both the SmMALA and SmHMC) is still computable.

Choice of the tensor metric $G(\cdot)$

- A natural choice for this metric is to take into account the local structure of the target distribution by using information from its hessian, i.e.

$$G(x_n) = -\Delta_{x_n}^{x_n} \log \widetilde{\pi}(x_n), \tag{15}$$

  where $\Delta_{x_n}^{x_n} := \nabla_{x_n} \nabla_{x_n}^T$ is the second derivative operator.
  $\Rightarrow$ Use of local curvature of the target distribution

- **one major issue** with this choice (unless the target distribution is log-concave) this negative Hessian will not be globally positive-definite
- [Betancourt, 2013] proposes a smooth absolute transformation of the eigenvalues that maps this negative Hessian metric into a positive-definite matrix in a way that the derivative of this transformed metric (required in both the SmMALA and SmHMC) is still computable.

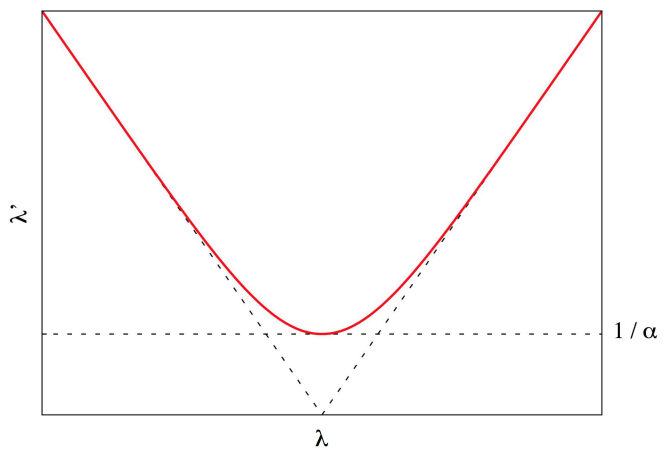
Choice of the tensor metric $G(\cdot)$

- A natural choice for this metric is to take into account the local structure of the target distribution by using information from its hessian, i.e.

$$G(x_n) = -\Delta_{x_n}^{x_n} \log \widetilde{\pi}(x_n), \tag{15}$$

where $\Delta_{x_n}^{x_n} := \nabla_{x_n} \nabla_{x_n}^T$ is the second derivative operator.
$\Rightarrow$ Use of local curvature of the target distribution

- **one major issue** with this choice (unless the target distribution is log-concave) this negative Hessian will not be globally positive-definite
- [Betancourt, 2013] proposes a smooth absolute transformation of the eigenvalues that maps this negative Hessian metric into a positive-definite matrix in a way that the derivative of this transformed metric (required in both the SmMALA and SmHMC) is still computable.



Choice of the tensor metric $G(\cdot)$

- An alternative strategy, used in [Girolami and Calderhead, 2011], consists in choosing $G(x_n)$ as a Fisher metric. In our context of filtering, this metric will be defined as :

$$G(x_n) = -\mathbb{E}_{Y_n|X_n}\left[\Delta^{x_n}_{x_n} \log g_n(y_n|x_n)\right] - \Delta^{x_n}_{x_n} \log f_n(x_n|x_{n-1} = X^j_{n,n-1}) \tag{16}$$

  which corresponds to the expectation over the data of the metric defined previously in Eq. (15)

  $\Rightarrow$ Positive-definite metric as long as the prior is log concave

- In this work, we propose a simpler strategy (when prior is not log concave) $\rightsquigarrow$ approximating the prior distribution with a multivariate normal distribution :

$$\begin{aligned} G(x_n) &= -\mathbb{E}_{Y_n|X_n}\left[\Delta^{x_n}_{x_n} \log g_n(y_n|x_n)\right] - \Delta^{x_n}_{x_n} \log \mathcal{N}(x_n; \widetilde{\mu}_n, \widetilde{\Sigma}_n), \\ &= -\mathbb{E}_{Y_n|X_n}\left[\Delta^{x_n}_{x_n} \log g_n(y_n|x_n)\right] + \widetilde{\Sigma}_n^{-1}, \end{aligned} \tag{17}$$

  where $\widetilde{\Sigma}_n = \mathrm{Var}_{f_n}\left(X_n|X^j_{n,n-1}\right)$ is the covariance matrix of $X_n|X^j_{n,n-1}$ from the true prior distribution.

Outline

Numerical Simulations : Model

- We consider a time-varying spatially dependent continuous process defined over a 2-dimensional space which is observed sequentially in time by d sensors deployed over a 2-D monitoring region.
- Each sensor therefore collects, independently of each other, at time n some noisy information about the phenomenon of interest at its specific location, i.e. $\forall k = 1, \ldots, d$:

$$Y_n(k)|X_n(k) = x_n(k) \sim g_n(y_n(k)|x_n(k)) \tag{18}$$

- To model the spatial and temporal dependencies we consider the following multivariate Generalized Hyperbolic (GH) distribution [McNeil et al., 2005] as prior distribution :

$$\begin{aligned} f_n(x_n|x_{n-1}) \propto & K_{\lambda-d/2}\left(\sqrt{(\chi+Q(x_n))(\psi+\gamma^T\Sigma^{-1}\gamma)}\right) \\ & \times \frac{e^{(x_n-\mu_n)^T\Sigma^{-1}\gamma}}{\sqrt{(\chi+Q(x_n))(\psi+\gamma^T\Sigma^{-1}\gamma)}^{\frac{d}{2}-\lambda}} \end{aligned} \tag{19}$$

where

- $Q(x_n) = (x_n-\mu_n)^T\Sigma^{-1}(x_n-\mu_n)$ and $\mu_n = \alpha x_{n-1} \in \mathbb{R}^d$ is the location parameter with $\alpha \in \mathbb{R}$
- K_λ : the modified Bessel function of the second kind of order λ
- λ, χ and ψ : shape of the distribution
- $\Sigma \in \mathbb{R}^{d\times d}$ is the dispersion matrix and the vector $\gamma \in \mathbb{R}^d$ is the skewness parameter.

Numerical Simulations : Model

- We consider a time-varying spatially dependent continuous process defined over a 2-dimensional space which is observed sequentially in time by d sensors deployed over a 2-D monitoring region.
- Each sensor therefore collects, independently of each other, at time n some noisy information about the phenomenon of interest at its specific location, i.e. $\forall k = 1, \ldots, d$:

$$Y_n(k)|X_n(k) = x_n(k) \sim g_n(y_n(k)|x_n(k)) \tag{18}$$

- To model the spatial and temporal dependencies we consider the following multivariate Generalized Hyperbolic (GH) distribution [McNeil et al., 2005] as prior distribution :

$$\begin{aligned} f_n(x_n|x_{n-1}) \propto & K_{\lambda-d/2}\left(\sqrt{(\chi + Q(x_n))(\psi + \gamma^T\Sigma^{-1}\gamma)}\right) \\ & \times \frac{e^{(x_n-\mu_n)^T\Sigma^{-1}\gamma}}{\sqrt{(\chi + Q(x_n))(\psi + \gamma^T\Sigma^{-1}\gamma)}^{\frac{d}{2}-\lambda}} \end{aligned} \tag{19}$$

where

- $Q(x_n) = (x_n - \mu_n)^T\Sigma^{-1}(x_n - \mu_n)$ and $\mu_n = \alpha x_{n-1} \in \mathbb{R}^d$ is the location parameter with $\alpha \in \mathbb{R}$
- K_λ : the modified Bessel function of the second kind of order λ
- λ, χ and ψ : shape of the distribution
- $\Sigma \in \mathbb{R}^{d\times d}$ is the dispersion matrix and the vector $\gamma \in \mathbb{R}^d$ is the skewness parameter.

Numerical Simulations : Model

- We consider a time-varying spatially dependent continuous process defined over a 2-dimensional space which is observed sequentially in time by d sensors deployed over a 2-D monitoring region.
- Each sensor therefore collects, independently of each other, at time n some noisy information about the phenomenon of interest at its specific location, i.e. $\forall k = 1, \ldots, d$:

$$Y_n(k)|X_n(k) = x_n(k) \sim g_n(y_n(k)|x_n(k)) \tag{18}$$

- To model the spatial and temporal dependencies we consider the following multivariate Generalized Hyperbolic (GH) distribution [McNeil et al., 2005] as prior distribution :

$$\begin{aligned} f_n(x_n|x_{n-1}) \propto & K_{\lambda-d/2}\left(\sqrt{(\chi + Q(x_n))(\psi + \gamma^T\Sigma^{-1}\gamma)}\right) \\ & \times \frac{e^{(x_n-\mu_n)^T\Sigma^{-1}\gamma}}{\sqrt{(\chi + Q(x_n))(\psi + \gamma^T\Sigma^{-1}\gamma)}^{\frac{d}{2}-\lambda}} \end{aligned} \tag{19}$$

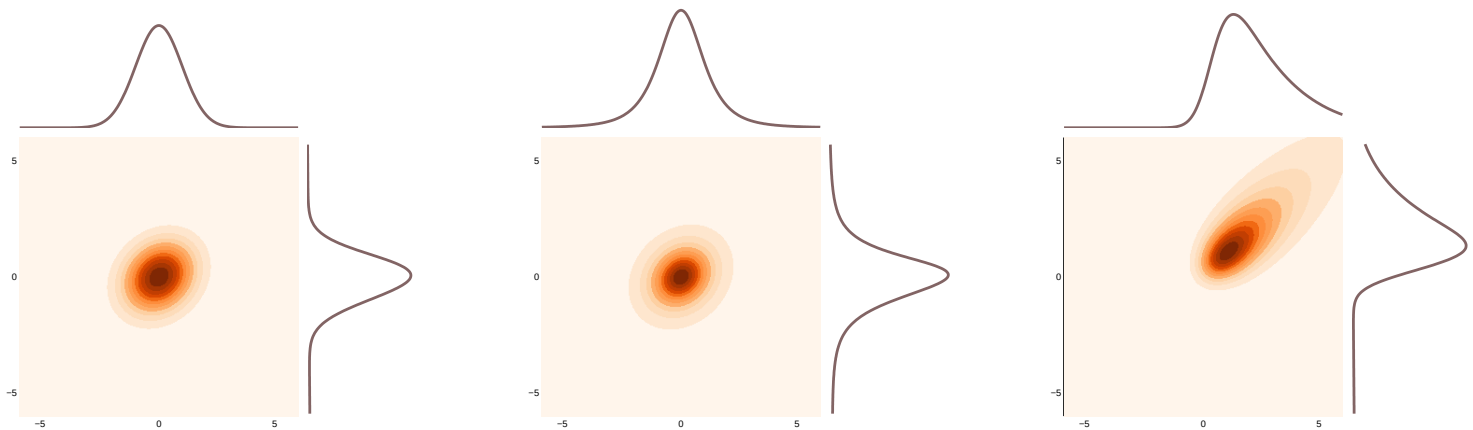
where

- $Q(x_n) = (x_n - \mu_n)^T\Sigma^{-1}(x_n - \mu_n)$ and $\mu_n = \alpha x_{n-1} \in \mathbb{R}^d$ is the location parameter with $\alpha \in \mathbb{R}$
- K_λ : the modified Bessel function of the second kind of order λ
- λ, χ and ψ : shape of the distribution
- $\Sigma \in \mathbb{R}^{d\times d}$ is the dispersion matrix and the vector $\gamma \in \mathbb{R}^d$ is the skewness parameter.

Numerical Simulations : Model

GH distribution : extremely flexible (heavy-tailed and asymmetric data)

$\rightsquigarrow$ many distributions are special case : normal, normal inverse Gaussian, skewed-t, etc



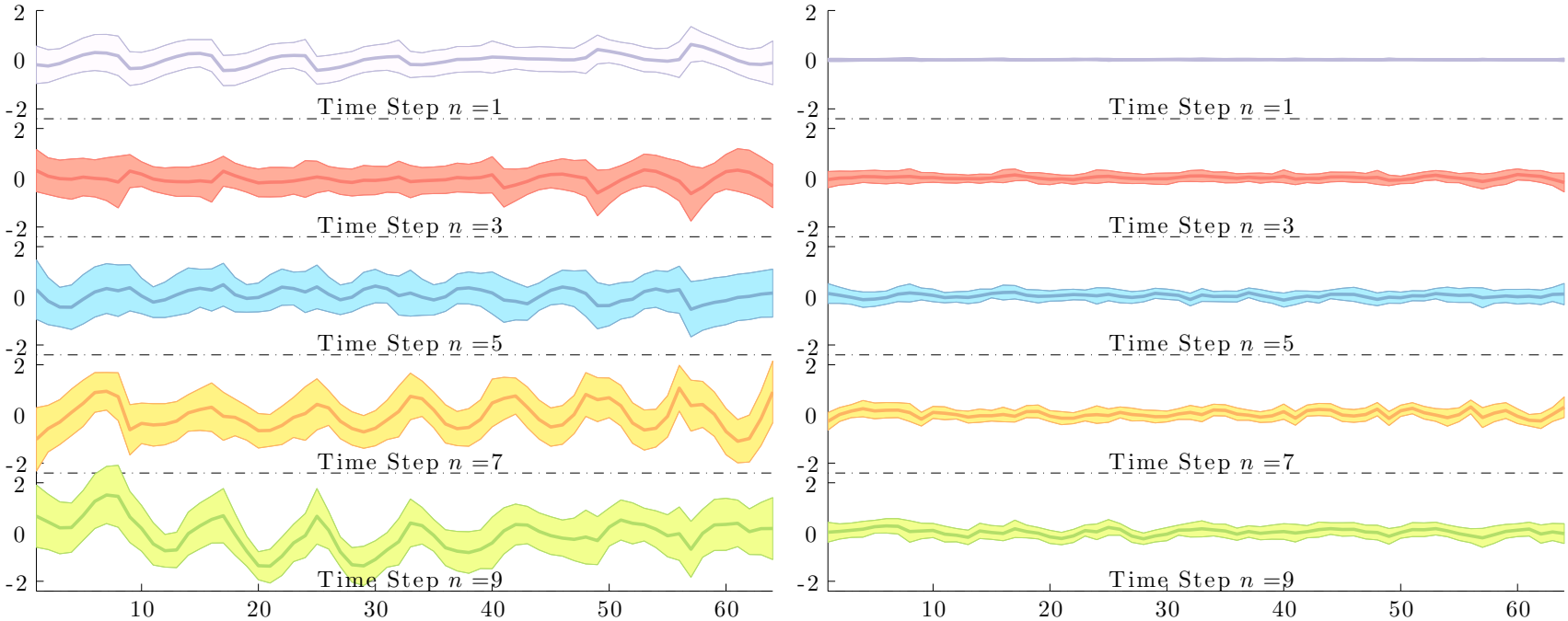
(a) (b) (c)

Figure: Illustration of few distributions from the Generalized Hyperbolic family with $[\Sigma]_{12} = 0.2$, $[\Sigma]_{11} = [\Sigma]_{22} = 1$, $\mu(1) = \mu(2) = 0$, $\lambda = -\nu/2$, $\chi = \nu$ and $\psi \to 0$. **(a)** : bivariate Normal distribution ($\gamma \to 0$ and $\nu \to \infty$) - **(b)** : Bivariate multivariate t distribution ($\gamma \to 0$ and $\nu = 3$) - **(c)** : Bivariate GH skewed-t distribution ($\gamma(1) = \gamma(2) = 2$ and $\nu = 3$)

As a first example, we consider the simplest special case : the multivariate normal distribution and a normal likelihood, i.e.

$$\begin{aligned} f_n(x_n|x_{n-1}) &= \mathcal{N}\left(x_n; \alpha x_{n-1}, \Sigma\right), \\ g_n(y_n|x_n) &= \mathcal{N}\left(y_n; x_n, \Sigma_y\right), \end{aligned} \tag{20}$$

Such a model is interesting for the understanding and the study of approximation methods since the posterior distribution can be derived analytically via the use of the Kalman filter [Kalman, 1960].



(a) **SIR** (b) **SmHMC**

Figure: Evolution of the mean ($\pm$ standard deviation) of the error between the posterior mean obtained by the different algorithms and the true one (obtained by using Kalman equations) across the $d = 64$ dimensions that composed the state and at different time step (results are obtained with 100 runs on the same data set - $N = 200$).

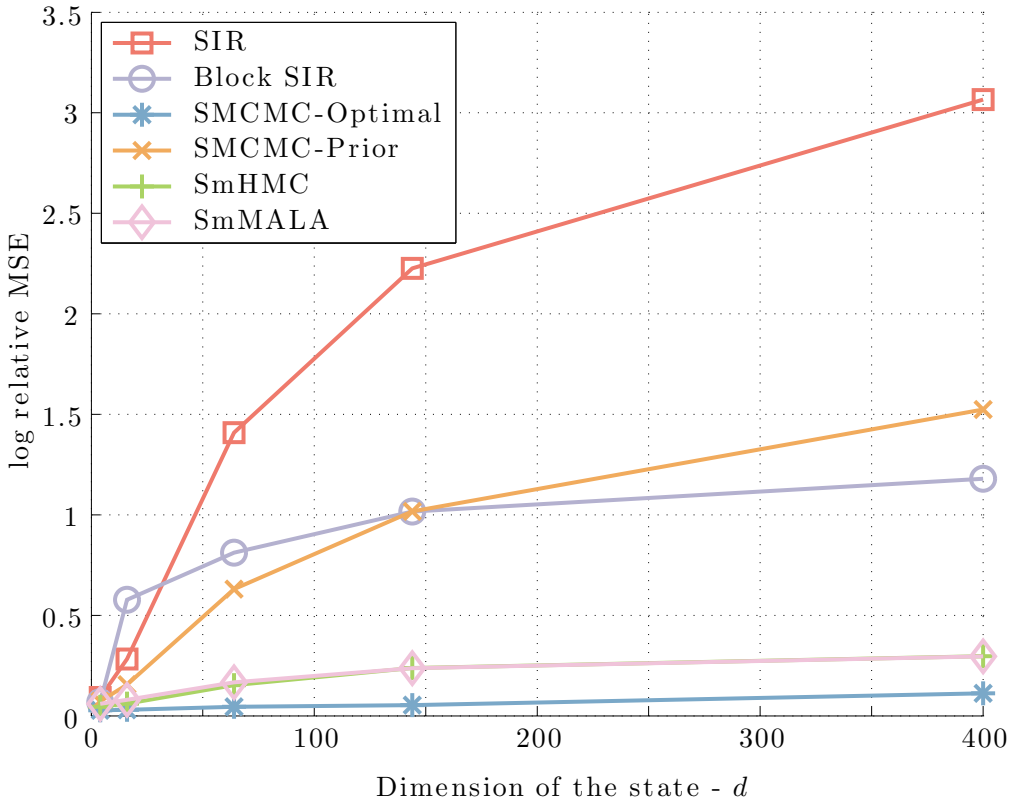


Figure: Log relative (to the optimal one given by Kalman equations) Mean squared error (average over time) for the different algorithms as the dimension of the state, d, increases. ($N = 200$).

Second Example

The likelihood is defined as

$$g_n(y_n|x_n) = \prod_{k=1}^{d} \mathcal{P}_o\left(y_n(k); m_1 \exp(m_2 x_n(k))\right) \tag{21}$$

The parameters of the prior is set to obtain the skewed-t process ($\lambda = -\nu/2$, $\chi = \nu$ and $\psi \to 0$)

Method	Dimension d		
	144	400	1024
SIR	4.95	8.87	12.17
Block SIR	1.29	1.48	1.55
SMCMC-Prior	1.68	3.35	5.23
Simplified SmMALA	0.61	0.79	0.91
SmMALA	0.60	0.76	0.88
SHMC	0.63	0.69	0.77
SmHMC	**0.55**	**0.58**	**0.65**

Table: Comparison of the mean squared error obtained on average at each sensor location with the different Monte-Carlo algorithms for several dimension d ($N = 200$).

	Method	Time [sec.]	ESS (Min., Med., Mean, Max.)	Mean ESS / Time
$d = 144$	SMCMC-Prior	11.4	(3, 8, 9, 31)	0.79
	Simplified SmMALA	**1.4**	(4, 13, 14, 32)	**10**
	SmMALA	5.7	(5, 17, 18, 35)	3.16
	SHMC	3.3	(7, 26, 33, 124)	**10**
	SmHMC	14.3	(**30**, **98**, **97**, **165**)	6.78
$d = 400$	SMCMC-Prior	194.5	(2, 5, 6, 27)	0.03
	Simplified SmMALA	**8.1**	(4, 10, 11, 32)	1.35
	SmMALA	26.2	(4, 11, 12, 34)	0.46
	SHMC	14.4	(4, 19, 20, 110)	1.39
	SmHMC	59.6	(**29**, **93**, **94**, **160**)	**1.58**

Table: Comparison of the different MCMC kernels in terms of Effective sample size (ESS) and computation time per time step ($N = 200$).

$$ESS = \frac{N}{1 + 2\sum_k \gamma(k)}$$

N : number of posterior samples (after the Burn-in period)
$\sum_k \gamma(k)$: sum of the K monotone sample autocorrelations as estimated by the initial monotone sequence estimator of [Geyer, 1992]

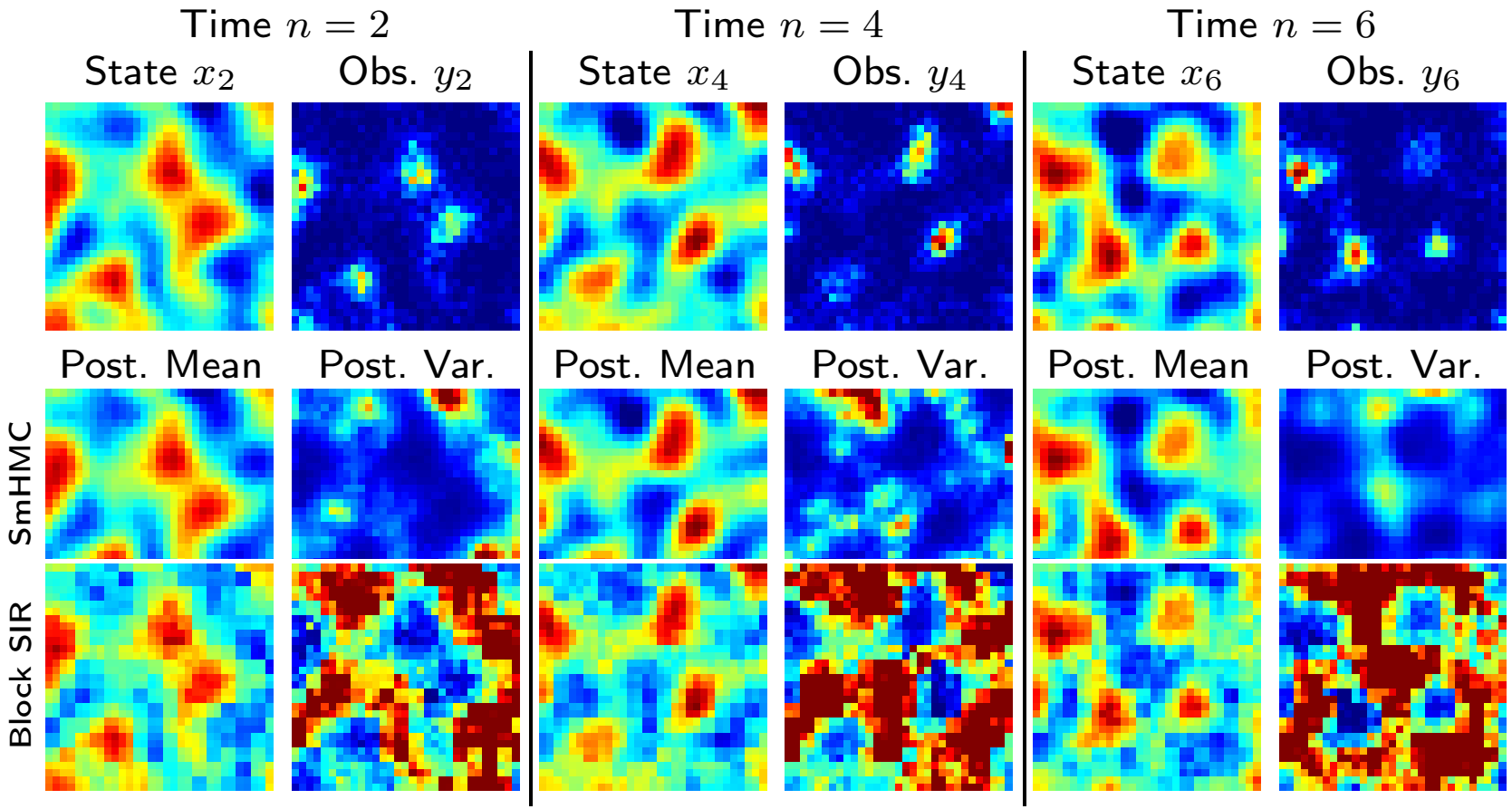


Figure: Illustration of the approximated posterior mean and variance at different time steps for several Monte-Carlo algorithms ($d = 1024$ - $N = 200$)

Outline

Conclusion

- Provide a unifying framework of the sequential Markov Chain Monte Carlo algorithms which constitute a promising alternative to traditional sequential Monte-Carlo methods,
- Discuss the choice of MCMC kernels to provide a useful guide for practitioners,
- Propose novel efficient kernels adapted to this SMCMC framework in order to increase the efficiency of such approaches when dealing with high-dimensional filtering problems,
- Demonstrate the significant gain that can be obtained with the SMCMC in a challenging high-dimensional filtering problem.

Bibliography

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.

Beskos, A., Crisan, D., Jasra, A., Kamatani, K. and Zhou, Y. (2014).

A Stable Particle Filter in High-Dimensions.
arXiv.org .

Betancourt, M. (2013).

A General Metric for Riemannian Manifold Hamiltonian Monte Carlo.
Geometric Science of Information, Lecture Notes in Computer Science, Springer ***8085***, 327–334.

Brockwell, A., Del Moral, P. and Doucet, A. (2010).

Sequentially interacting Markov chain Monte Carlo methods.
Ann. Stat. ***38***, 3387–3411.

Duane, S., Kennedy, A. D., Pendleton, B. J. and Roweth, D. (1987).

Hybrid Monte Carlo.
Physics Letters B ***195***, 216–222.

Ermak, D. L. (1975).

A computer simulation of charged particles in solution. I. Technique and equilibrium properties.
J .Chem. Phys. ***62***, 4189–4196.

Geyer, C. (1992).

Practical Markov Chain Monte Carlo (with discussion).
Statistical Science ***7***, 473–511.

Girolami, M. and Calderhead, B. (2011).

Riemann manifold Langevin and Hamiltonian Monte Carlo methods.
J. R. Stat. Soc. Series B Stat. Methodol. ***73***, 1–37.

Golightly, A. and Wilkinson, D. (2006).

Bayesian sequential inference for nonlinear multivariate diffusions.
Stat. and Comput. ***16***, 323–338.

Green, P. J., Latuszynski, K., Pereyra, M. and Robert, C. P. (2015).

Bayesian computation : a perspective on the current state, and sampling backwards and forwards.
arXiv.org .

Kalman, R. E. (1960).

A New Approach to Linear Filtering and Prediction Problems.
Transactions of the ASME–Journal of Basic Engineering ***82***, **35–45**.

Livingstone, S. and Girolami, M. (2014).

Information-geometric Markov Chain Monte Carlo methods using Diffusions.
arXiv.org .

McNeil, A. J., Frey, R. and Embrechts, P. (2005).

Quantitative Risk Management : Concepts, Techniques, and Tools.
Princeton University Press.

Neal, R. (2010).

MCMC using Hamiltonian dynamics.
In Handbook of Markov Chain Monte Carlo, **(Brooks, S., Gelman, A., Jones, G. and Meng, X.-L., eds)**,. **Chapman & Hall / CRC Press**.

Rebeschini, P. and van Handel, R. (2015).

Can local particle filters beat the curse of dimensionality ?
Ann. Appl. Probab. .

Roberts, G. and Stramer, O. (2002).

Langevin Diffusions and Metropolis-Hastings Algorithms.
Methodol. Comput. Appl. Probab. ***4***, **337–357**.

Rossky, P. J., Doll, J. D. and Friedman, H. L. (1978).

Brownian dynamics as smart Monte Carlo simulation.
J .Chem. Phys. ***69***, **4628**.

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**.

Snyder, C., Bengtsson, T., Bickel, P. and Anderson, J. (2008).

Obstacles to high-dimensional particle filtering.

Monthly Weather Review *136*, 4629–4640.

Xifara, T., Sherlock, C., Livingstone, S., Byrne, S. and Girolami, M. (2014).
Langevin diffusions and the Metropolis-adjusted Langevin algorithm.
Stat. and Comput. *91*, 14–19.