Particle Markov chain Monte Carlo methods

Meng Li Corey Smith

The Ohio State University

December 7, 2015

Overview

Introduction

- Markov chain Monte Carlo (MCMC) methods have become a standard tool for statisticians.
- Methods like the Metropolis-Hastings algorithm are commonly used to sample from high dimensional distributions that may also exhibit complex dependency structures.
- The efficiency of these algorithms depends on an appropriate choice for the proposal distribution.
- A good proposal distribution should capture important characteristics of the target distribution, such as its scale and dependence structure.

Introduction

Goal:

- Design an efficient proposal distribution.

Problem:

- This is a feasible task in small dimensions, but it can be very difficult in high dimensions.

Existing Solutions:

- Focus on subcomponents of the target distribution; this can ignore some of the dependency that exists!

Particle MCMC Solution:

- Use sequential Monte Carlo (SMC) and MCMC methods together to design efficient MCMC algorithms with little user design.

Overview

Motivation for SMC: Hidden Markov Model

In this model, we have a hidden Markov process $\{X_n; n \geq 1\}$ that is characterized some initial density $X_1 \sim \mu_\theta(\cdot)$ and transition probability density

$$X_{n+1}|X_n = x \sim f_\theta(\cdot|x)$$

for some $\theta \in \Theta$.

The process $\{X_n\}$ is observed through another process $\{Y_n; n \geq 1\}$. The observations (the Y_i's) are assumed to be conditionally independent given $\{X_n\}$ with densities of the form

$$Y_n|X_1, \ldots, X_n = x, \ldots, X_m \sim g_\theta(\cdot|x), \forall n \in \{1, \ldots, m\}.$$

Motivation for SMC: Hidden Markov Model

Goal:
Perform Bayesian inference conditional on the observations $y_{1:T} = (y_1, \ldots, y_T)$ for some $T \geq 1$.

This will require the posterior density $p_\theta(x_{1:T}|y_{1:T})$.

If $\theta \in \Theta$ is known, the posterior is proportional to

$$p_\theta(x_{1:T}, y_{1:T}) = \mu_\theta(x_1) g_\theta(y_1|x_1) \prod_{n=2}^{T} f_\theta(x_n|x_{n-1}) g_\theta(y_n|x_n).$$

If $\theta \in \Theta$ is unknown, the posterior is proportional to

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

Background for SMC: Importance Sampling

Used to sample from a target distribution p by using an importance density q.

1. Sample candidates $Y_1, \ldots, Y_m$ i.i.d. from q.
2. Calculate the (normalized) importance weights, $W(Y_1), \ldots, W(Y_m)$ defined by

$$W(y_i) = \frac{p(y_i)/q(y_i)}{\sum_{i=1}^{m} p(y_i)/q(y_i)}.$$

3. Resample $X_1, \ldots, X_n$ from $Y_1, \ldots, Y_m$ (with replacement) with probabilities $W(Y_1), \ldots, W(Y_m)$.

The sample $X_1, \ldots, X_n$ has distribution that converges to p as $m \to \infty$.

Sequential Monte Carlo Algorithm (HMM)

SMC methods yield an estimate, $\hat{p}$, for the posterior described previously.

At time 1, we approximate $p_\theta(x_1|y_1)$ using importance sampling with an importance density $q_\theta(x_1|y_1)$.

N particles $\{X_1^k\} = (X_1^1, \ldots, X_1^N)$ are sampled from $q_\theta(x_1|y_1)$ and (normalized) importance weights, $\{W_1^k\} = (W_1^1, \ldots, W_1^N)$, are calculated.

Using these particles and weights, we can (re)sample N particles (approximately distributed as $p_\theta(x_1|y_1)$) by sampling from our estimate of p given by

$$\hat{p}_\theta(x_1|y_1) = \sum_{k=1}^{N} W_1^k \delta_{X_1^k}(x_1).$$

Sequential Monte Carlo Algorithm (HMM)

At time 2, we again use importance sampling to approximate $p_\theta(x_{1:2}|y_{1:2})$. We reuse the N samples (particles) from time 1 and extend each particle with an importance sampling density $q_\theta(x_2|y_2, x_1)$.

This yields samples that are approximately distributed as

$$p_\theta(x_1|y_1)q_\theta(x_2|y_2, x_1).$$

Importance weights $\{W_2^k\}$ are recalculated since our target is $p_\theta(x_{1:2}|y_{1:2})$. We then resample N particles from

$$\hat{p}_\theta(dx_{1:2}|y_{1:2}) = \sum_{k=1}^{N} W_2^k \delta_{X_{1:2}^k}(dx_{1:2}).$$

Repeat until time T.

Sequential Monte Carlo Algorithm (HMM)

At time T, this algorithm yields an approximation of the joint posterior density $p_\theta(dx_{1:T}|y_{1:T})$ given by

$$\hat{p}_\theta(dx_{1:T}|y_{1:T}) = \sum_{k=1}^{N} W_T^k \delta_{X_{1:T}^k}(dx_{1:T}).$$

This algorithm also provides us with an estimate of the marginal likelihood $p_\theta(y_{1:T})$ given by

$$\hat{p}_\theta(y_{1:T}) = \hat{p}_\theta(y_1) \prod_{n=2}^{T} \hat{p}_\theta(y_n|y_{1:(n-1)})$$

where

$$\hat{p}_\theta(y_n|y_{1:(n-1)}) = \frac{1}{N} \sum_{k=1}^{N} w_n(X_{1:n}^k).$$

Sequential Monte Carlo: Design and Limitations

- This is a powerful algorithm because it requires very little user input, yet still provides useful results.
- We need only specify one-dimensional importance densities $q_\theta(x_1|y_1)$ and $q_\theta(x_n|y_n, x_{n-1})$ for $n \geq 2$.
- The authors suggest using $q_\theta(x_1|y_1) = \mu_\theta(x_1)$ and $q_\theta(x_n|y_n, x_{n-1}) = f_\theta(x_n|x_{n-1})$ for $n \geq 2$.
- When T is too large, we run into issues of degeneracy.
- Successive resampling diminishes the number of distinct values for x_n.

An Illustrative Example: Linear Gaussian SSM

- Linear Gaussian State-Space Model
- State process $\{X_t : t \in Z\}$:

$$X_t = \phi X_{t-1} + \epsilon_t$$

  where $\epsilon_t \overset{iid}{\sim} N(0, \sigma^2)$ and $|\phi| < 1$.
- Data process $\{Y_t : t \in Z\}$:

$$Y_t = \theta X_t + r_t$$

  where $r_t \overset{iid}{\sim} N(0, \gamma^2)$.
- Parameter vector: $\boldsymbol{\theta} = (\phi, \theta, \sigma^2, \gamma^2)$.

An Illustrative Example: Linear Gaussian SSM

- Suppose that the parameters $\boldsymbol{\theta} = (0.7, 1, 0.5, 0.1)$ are given.
- The goal is to estimate the system state $\{X_t : 1, 2, \ldots T\}$ conditioning on the observed data $Y_{1:T}$, that is

$$P_\theta(X_t \mid Y_{1:T} = y_{1:T}) \quad \text{for} \quad t = 1, \ldots T$$

1. Obtain $P_\theta(X_t \mid Y_{1:T} = y_{1:T}), \forall t = 1, \ldots T$ from Kalman Filter.
2. Obtain approximate samples from $P_\theta(X_t \mid Y_{1:T} = y_{1:T})$, $\forall t = 1, \ldots T$ using SMC.

SSM Example: Kalman Filter

Filtering: for $t = 1 \dots T$,

1. get forecasts $x_{t|t-1}$ and $P_{t|t-1}$ from

$$x_{t|t-1} = \phi x_{t-1|t-1}, \quad P_{t|t-1} = \sigma^2 + \phi^2 P_{t-1|t-1}$$

2. get analysis $x_{t|t}$ and $P_{t|t}$ from

$$K_t = \frac{\theta P_{t|t-1}}{\gamma^2 + \theta^2 P_{t|t-1}}$$
$$x_{t|t} = x_{t|t-1} + K_t(y_t - \theta x_{t|t-1})$$
$$P_{t|t} = (1 - K_t\theta)P_{t|t-1}$$

SSM Example: Kalman Filter

Smoothing: for $t = T - 1 \ldots 1$,

1. Obtain J_t from

$$J_t = \frac{\phi P_{t|t}}{P_{t+1|t}}$$

2. Obtain $x_{t|T}$ from

$$x_{t|T} = x_{t|t} + J_t(x_{t+1|T} - x_{t+1|t})$$

3. Obtain $P_{t|T}$ from

$$P_{t|T} = P_{t|t} + J_t^2(P_{t+1|T} - P_{t+1|t})$$

SSM Example: Sequential Monto Carlo

Step 1: $n = 1$

(1) Sample X_1^k from an arbitrary initial distribution for $X_1 \sim N(0, 5)$;
(2) Importance weights w_1 is

$$w_1(X_1^k) = p_\theta(y_1 \mid x_1)$$

(3) Normalized importance weight W_1 is obtained as

$$W_1(X_1^k) = \frac{w_1(X_1^k)}{\sum_{k=1}^{N} w_1(X_1^k)}$$

SSM Example: Sequential Monte Carlo

Step 2: $n = 2 \dots T$,

(1) Sample $A_{n-1}^k \sim Multinominal(\cdot \mid W_{n-1})$
(2) Sample $X_n^k \sim Normal(\phi X_{n-1}^{A_{n-1}^k}, \sigma^2)$ and set $X_{1:n}^k = (X_{1:n-1}^{A_{n-1}^k}, X_n^k)$
(3) Importance weights are calculated from

$$w_n(X_{1:n}^k) = p_\theta(y_n \mid x_n^k)$$

$$W_n(X_{1:n}^k) = \frac{w_n(X_{1:n}^k)}{\sum_{k=1}^{N} w_n(X_{1:n}^k)}$$

(4) $\hat{p}_\theta(y_n | y_{1:(n-1)}) = \frac{1}{N} \sum_{k=1}^{N} w_n(X_{1:n}^k)$

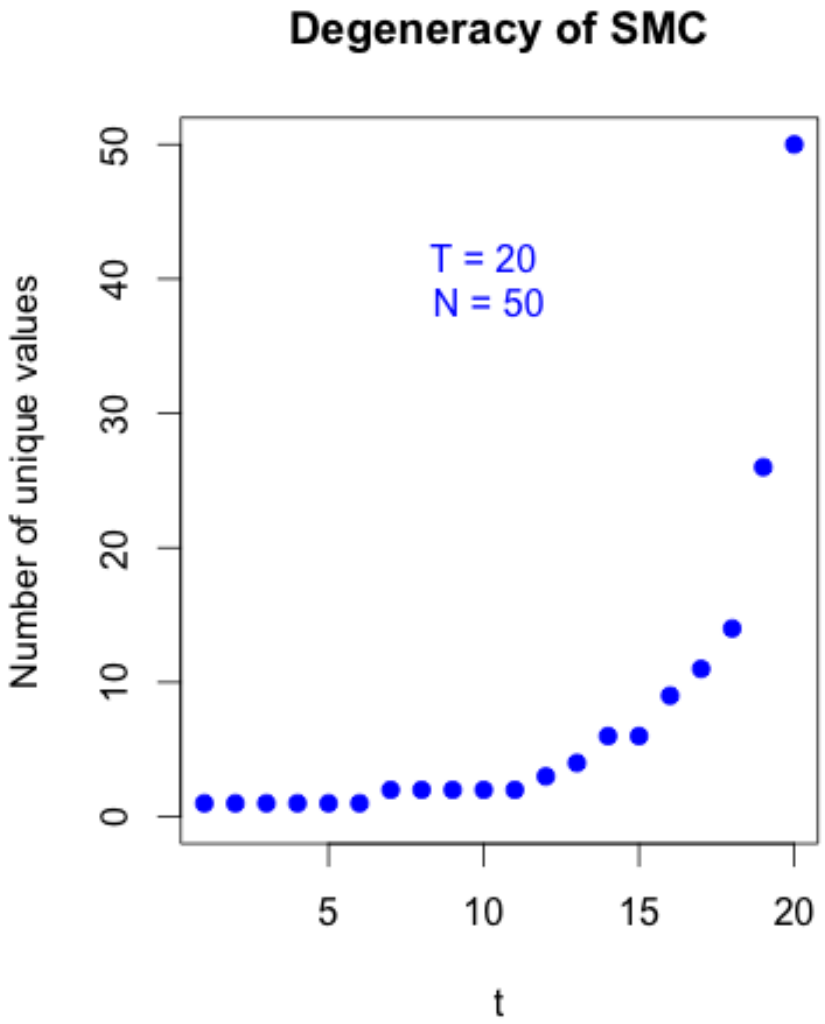


Figure: Number of unique values at time t

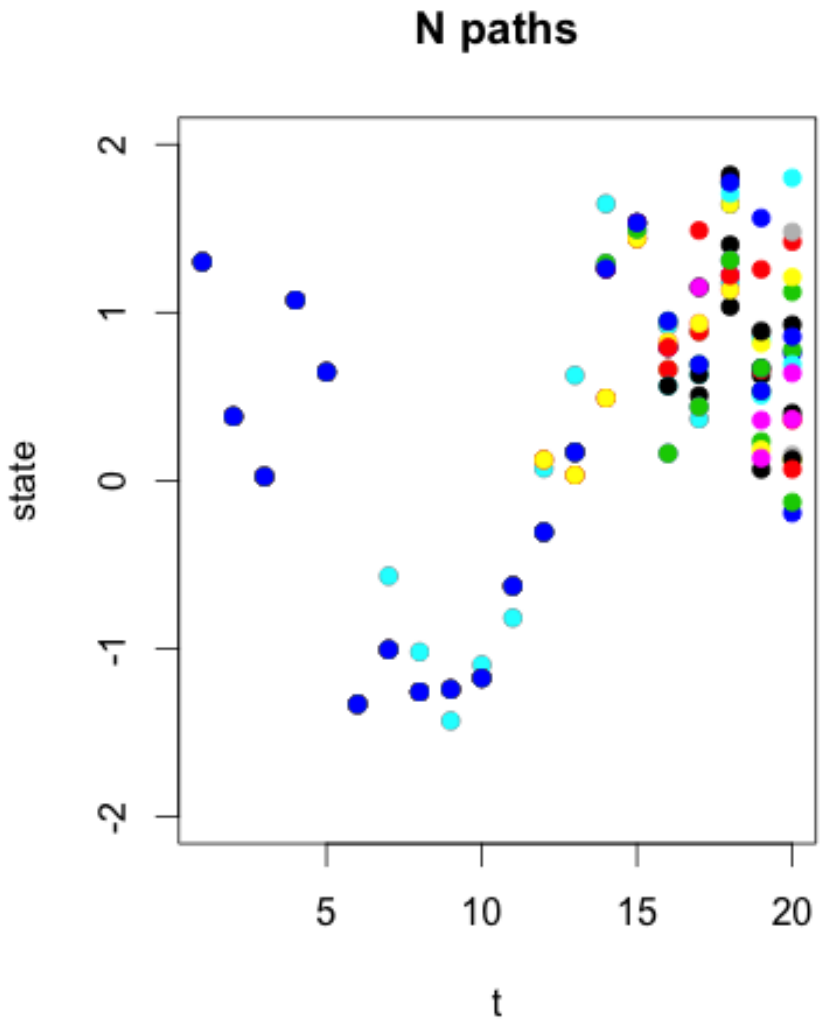


Figure: N Particles from SMC

Overview

Particle Independent Metropolis-Hastings (PIMH)

PIMH Setup/Notation:

- Target Density: $p_\theta(x_{1:T}|y_{1:T})$.
- Optimal Proposal Density: $q_\theta(x_{1:T}|y_{1:T}) = p_\theta(x_{1:T}|y_{1:T})$.
- Realistic Proposal Density: $q_\theta(x_{1:T}|y_{1:T}) = \hat{p}_\theta(x_{1:T}|y_{1:T})$.
- Implementing $\hat{p}_\theta$ into the Metropolis-Hastings algorithm yields a relatively simple and familiar method.

Particle MCMC

- SMC is an approximate simulation procedure for the target density $p_\theta(X_{1:T} \mid Y_{1:T})$.
- The output of an SMC algorithm targeting $p_\theta(X_{1:T} \mid Y_{1:T})$ using $N \geq 1$ particles is used as the proposal distribution for the usual MCMC algorithm.
- This cannot be implemented directly, as the evaluation of the acceptance ratio in MCMC requires the marginal density of a particle that is generated from an SMC algorithm.

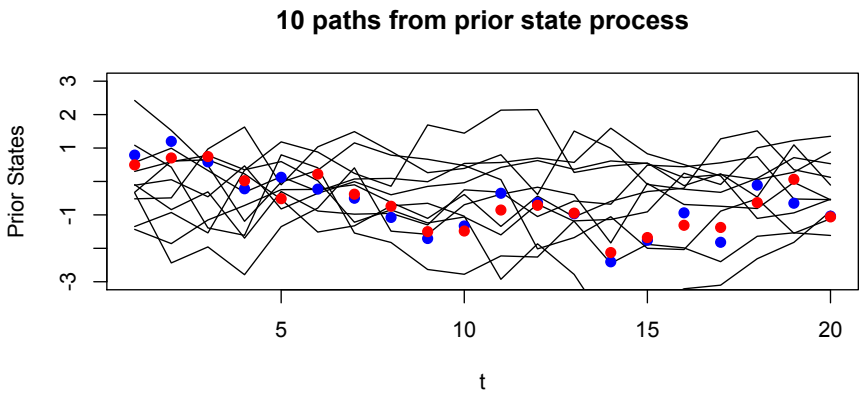
Particle Independent Metropolis-Hastings (PIMH)

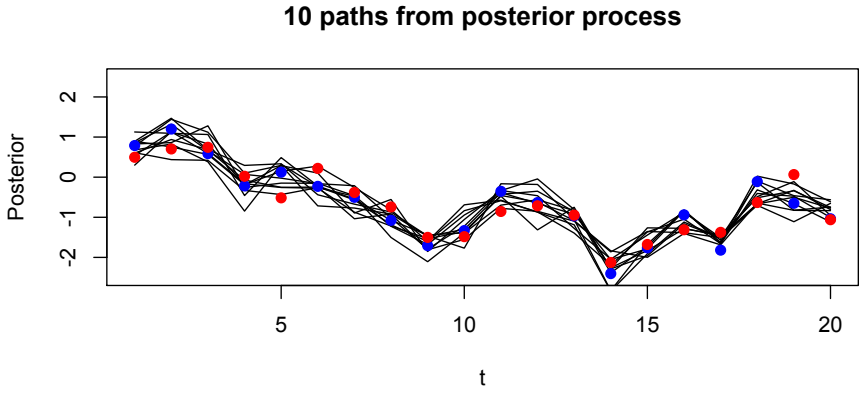
Step 1: iteration $i = 0$, run an SMC algorithm targeting $p_\theta(x_{1:T}|y_{1:T})$, sample $X_{1:T}(0) \sim \hat{p}_\theta(\cdot|y_{1:T})$, and let $\hat{p}_\theta(y_{1:T})(0)$ denote the corresponding marginal likelihood estimate.
Step 2: iteration $i \geq 1$,

1. run an SMC algorithm targeting $p_\theta(x_{1:T}|y_{1:T})$, sample $X^*_{1:T} \sim \hat{p}_\theta(\cdot|y_{1:T})$, and let $\hat{p}_\theta(y_{1:T})^*$ denote the corresponding marginal likelihood estimate, then
2. with probability

$$\min\left\{1, \frac{\hat{p}_\theta(y_{1:T})^*}{\hat{p}_\theta(y_{1:T})(i-1)}\right\}$$

set $X_{1:T}(i) = X^*_{1:T}$ and $\hat{p}_\theta(y_{1:T})(i) = \hat{p}_\theta(y_{1:T})^*$; else set $X_{1:T}(i) = X_{1:T}(i-1)$ and $\hat{p}_\theta(y_{1:T})(i) = \hat{p}_\theta(y_{1:T})(i-1)$


10 paths from prior state process
Prior States
3
2
1
-1
-3
5
10
15
20
t


10 paths from posterior process
Posterior
2
1
0
-1
-2
5
10
15
20
t

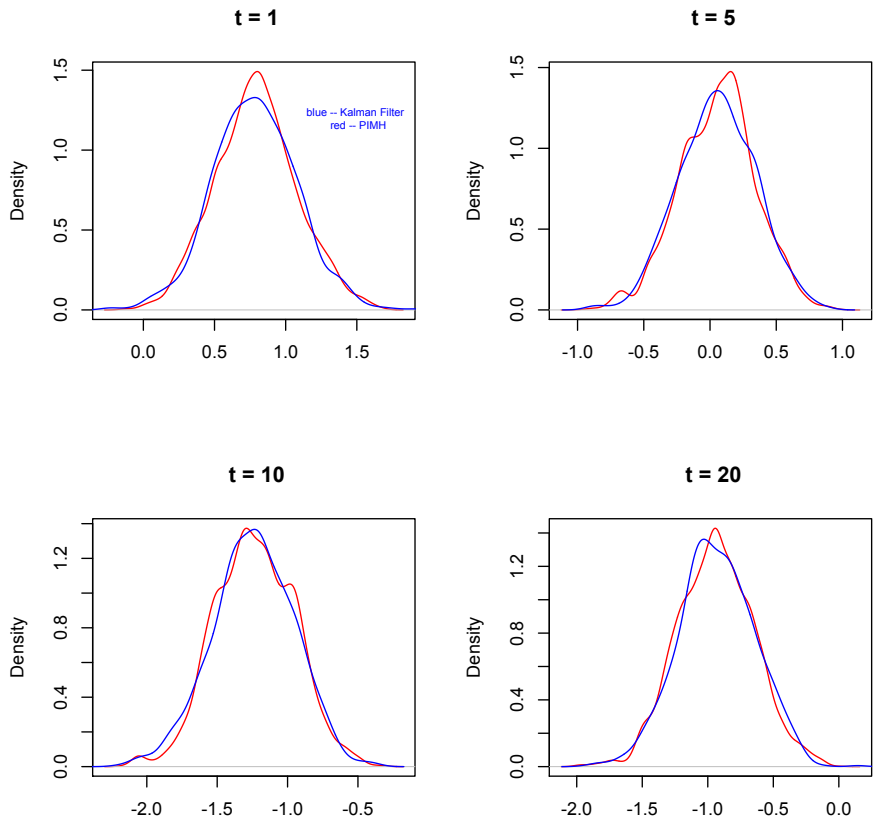


Figure: Marginal density plot of x_t given $y_{1:T}$

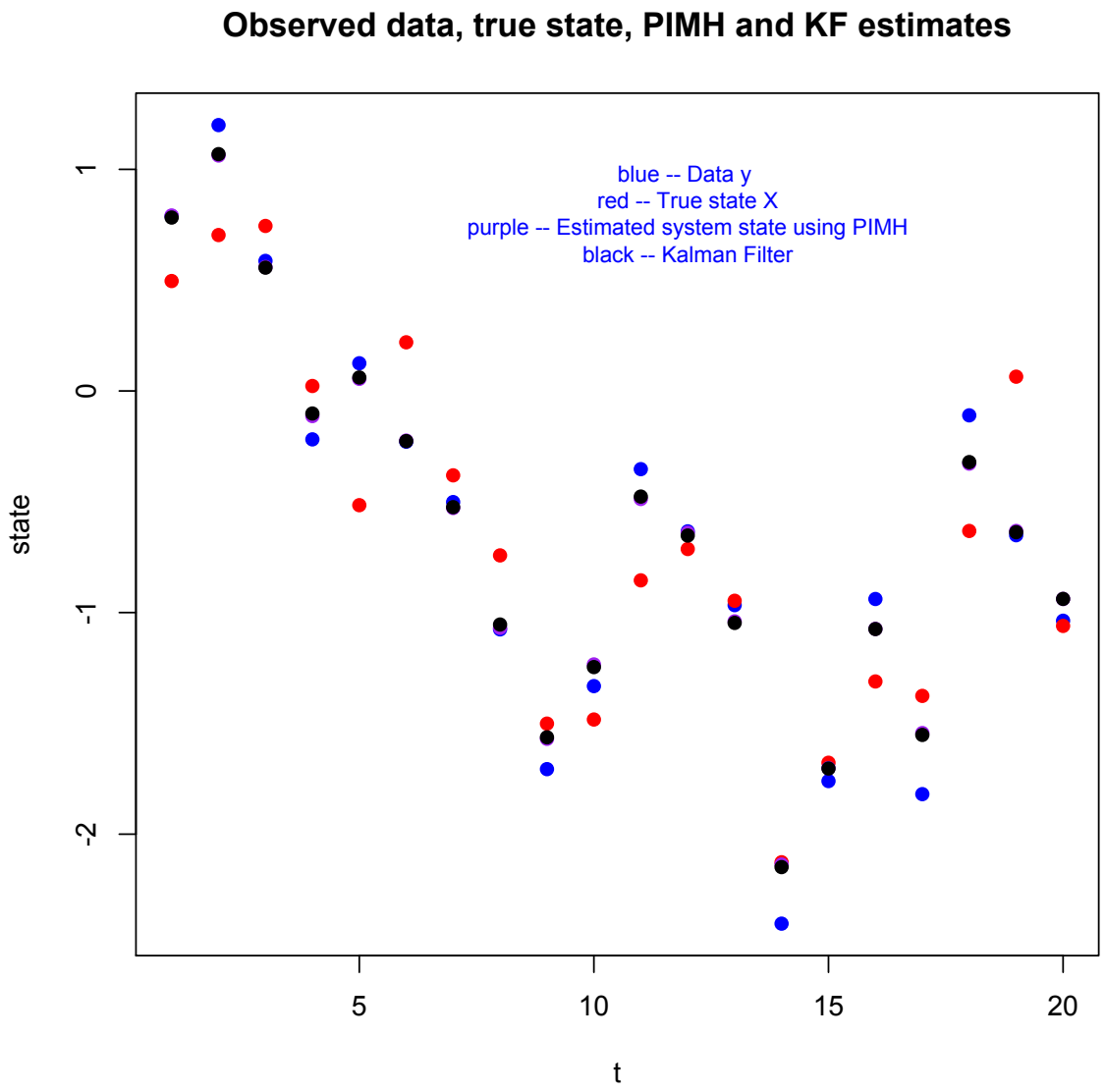


Figure: Plots of observed data and states

Particle MCMC with unknown θ

When the parameter θ is unknown and we are interested in sampling from $p(\theta, x_{1:T}|y_{1:T})$,

1. Particle Marginal Metropolis-Hastings Sampling(PMMH)
2. Particle Gibbs Sampling.

Particle Marginal Metropolis-Hastings Sampling(PMMH)

- Given that $p(\theta, x_{1:T}|y_{1:T}) = p(\theta \mid y_{1:T})p_\theta(x_{1:T} \mid y_{1:T})$, a natural choice of proposal density for an MH update is,

$$q(\theta^\star, x^\star_{1:T} \mid \theta, y_{1:T}) = q(\theta^\star \mid \theta)p_{\theta^\star}(x^\star_{1:T} \mid y_{1:T})$$

- The resulting MH acceptance ratio is given by

$$\frac{p(\theta^\star, x^\star_{1:T}) \mid y_{1:T})q(\theta, x^\star_{1:T} \mid \theta^\star, y_{1:T})}{p(\theta, x^\star_{1:T}) \mid y_{1:T}q(\theta^\star, x^\star_{1:T} \mid \theta, y_{1:T})}$$
$$= \frac{p^\star_\theta(y_{1:T})p(\theta^\star)q(\theta \mid \theta^\star)}{p_\theta(y_{1:T})p(\theta)q(\theta^\star \mid \theta)}$$

Particle Marginal Metropolis-Hastings Sampling(PMMH)

Step 1: at $i = 0$, set $\theta(0)$ arbitrarily and run an SMC algorithm targeting $p_{\theta(0)}(x_{1:T} \mid y_{1:T})$, sample $X_{1:T}(0) \sim \hat{p}_{\theta(0)}(\cdot|y_{1:T})$, and let $\hat{p}_{\theta(0)}(y_{1:T})$ denote the corresponding marginal likelihood estimate.
Step 2: iteration $i \geq 1$,

1. sample $\theta^\star \sim q(\cdot \mid \theta(i-1))$,
2. sample $X^*_{1:T} \sim \hat{p}_{\theta^\star}(\cdot|y_{1:T})$, and let $\hat{p}_{\theta^\star}(y_{1:T})$ denote the corresponding marginal likelihood estimate, then
3. with probability

$$\min\left\{1, \frac{\hat{p}^\star_\theta(y_{1:T})p(\theta^\star)q(\theta(i-1) \mid \theta^\star)}{\hat{p}_{\theta(i-1)}(y_{1:T})p(\theta(i-1))q(\theta^\star \mid \theta(i-1))}\right\}$$

set $\theta(i) = \theta^\star, X_{1:T}(i) = X^\star_{1:T}$ and $\hat{p}_{\theta(i)}(y_{1:T})(i) = \hat{p}_\theta(y_{1:T})^*$; otherwise set $\theta(i) = \theta(i-1), X_{1:T}(i) = X_{1:T}(i-1)$ and $\hat{p}_{\theta(i)}(y_{1:T}) = \hat{p}_{\theta(i-1)}(y_{1:T})$

Overview

Conclusions

1. In high dimensions, designing an efficient MCMC algorithm (proposal) "by hand" is not possible.

2. Breaking up the problem into many low dimensional problems often fails to include some information about the target density.

3. Combining sequential Monte Carlo methods with existing MCMC algorithms breaks the high dimensional problem in many low dimensional problems while still accounting for properties of the target distribution.

4. Particle MCMC methods provide efficient algorithms with little user design needed.

References

- Andrieu and A. Doucet and R. Holenstein (2010), Particle Markov chain Monte Carlo methods, Journal of the Royal Statistical Society, Series B, 72 269-342.
- C. K. Wikle and L. M. Berliner (2007), A Bayesian tutorial for data assimilation, Physica D, 230, 1-16.