BiiPS

a software for **B**ayesian **i**nference with **i**nteracting **P**article **S**ystems

A. Todeschini, F. Caron, P. Legrand, P. Del Moral

Summary

1. Context
2. BiiPS software
3. Example in financial econometry

Summary

1. Context
2. BiiPS software
3. Example in financial econometry

Context

- Bayesian inference: we want to sample according to the posterior distribution $p(X|Y)$

$$p(X|Y) = \frac{p(X, Y)}{p(Y)}$$

- Many people use MCMC methods with BUGS software
  - Provides a, so-called, BUGS language for describing a graphical model
  - Expert system drives MCMC methods (Gibbs, Slice, Metropolis, ...) in a 'black-box' fashion
  - Very popular among practitioners, applying MCMC methods to a wide range of applications
- Having such a 'black-box' software (generic and easy to use) for SMC methods would be great
- The *BiiPS* project have been trying to bridge this gap for 3 years

Context

- Bayesian inference: we want to sample according to the posterior distribution $p(X|Y)$

$$p(X|Y) = \frac{p(X, Y)}{p(Y)}$$

- Many people use MCMC methods with BUGS software
  - Provides a, so-called, BUGS language for describing a graphical model
  - Expert system drives MCMC methods (Gibbs, Slice, Metropolis, ...) in a 'black-box' fashion
  - Very popular among practitioners, applying MCMC methods to a wide range of applications
- Having such a 'black-box' software (generic and easy to use) for SMC methods would be great
- The *BiiPS* project have been trying to bridge this gap for 3 years

Context

- Bayesian inference: we want to sample according to the posterior distribution $p(X|Y)$

$$p(X|Y) = \frac{p(X,Y)}{p(Y)}$$

- Many people use MCMC methods with BUGS software
  - Provides a, so-called, BUGS language for describing a graphical model
  - Expert system drives MCMC methods (Gibbs, Slice, Metropolis, ...) in a 'black-box' fashion
  - Very popular among practitioners, applying MCMC methods to a wide range of applications
- Having such a 'black-box' software (generic and easy to use) for SMC methods would be great
- The *BiiPS* project have been trying to bridge this gap for 3 years

Context

- Bayesian inference: we want to sample according to the posterior distribution $p(X|Y)$

$$p(X|Y) = \frac{p(X,Y)}{p(Y)}$$

- Many people use MCMC methods with BUGS software
  - Provides a, so-called, BUGS language for describing a graphical model
  - Expert system drives MCMC methods (Gibbs, Slice, Metropolis, ...) in a 'black-box' fashion
  - Very popular among practitioners, applying MCMC methods to a wide range of applications
- Having such a 'black-box' software (generic and easy to use) for SMC methods would be great
- The *BiiPS* project have been trying to bridge this gap for 3 years

Graphical Models / Bayesian networks

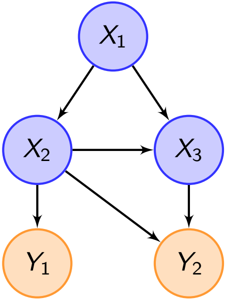


Figure : Directed acyclic graph

The graph displays a factorization of the joint distribution:

$p(x_{1:3}, y_{1:2})$

Graphical Models / Bayesian networks

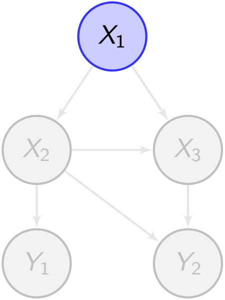


Figure : Directed acyclic graph

The graph displays a factorization of the joint distribution:

$$p(x_{1:3}, y_{1:2}) = p(x_1)\ p(x_2|x_1)\ p(y_1|x_2)\ p(x_3|x_1, x_2)\ p(y_2|x_2, x_3)$$

Graphical Models / Bayesian networks

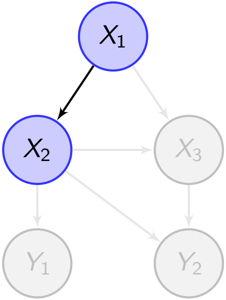


Figure : Directed acyclic graph

The graph displays a factorization of the joint distribution:

$$p(x_{1:3}, y_{1:2}) = p(x_1)\ p(x_2|x_1)\ p(y_1|x_2)\ p(x_3|x_1, x_2)\ p(y_2|x_2, x_3)$$

Graphical Models / Bayesian networks

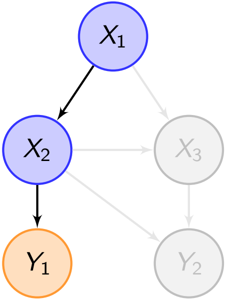


Figure : Directed acyclic graph

The graph displays a factorization of the joint distribution:

$$p(x_{1:3}, y_{1:2}) = p(x_1)\ p(x_2|x_1)\ p(y_1|x_2)$$
$$p(x_3|x_1, x_2)\ p(y_2|x_2, x_3)$$

Graphical Models / Bayesian networks

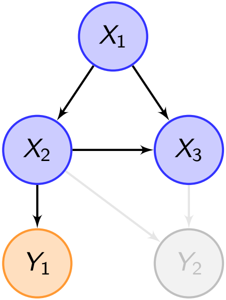


Figure : Directed acyclic graph

The graph displays a factorization of the joint distribution:

$$p(x_{1:3}, y_{1:2}) = p(x_1)\ p(x_2|x_1)\ p(y_1|x_2)\ p(x_3|x_1, x_2)\ p(y_2|x_2, x_3)$$

Graphical Models / Bayesian networks

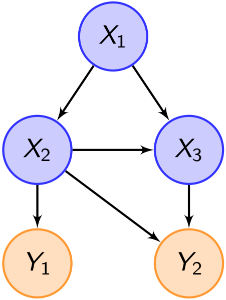


Figure : Directed acyclic graph

The graph displays a factorization of the joint distribution:

$$p(x_{1:3}, y_{1:2}) = p(x_1)\ p(x_2|x_1)\ p(y_1|x_2)\ p(x_3|x_1, x_2)\ p(y_2|x_2, x_3)$$

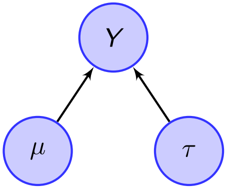
BUGS language

- S-like declarative language for describing graphical models
- Stochastic relations
- Deterministic relations

BUGS language

- S-like declarative language for describing graphical models
- Stochastic relations
- Deterministic relations

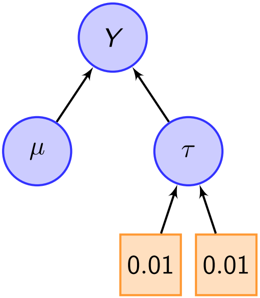
```
Y ~ dnorm(mu, tau)
```



BUGS language

- S-like declarative language for describing graphical models
- Stochastic relations
- Deterministic relations

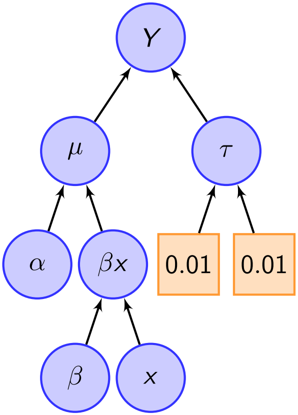
```
Y ~ dnorm(mu, tau)
tau ~ dgamma(0.01, 0.01)
```



BUGS language

- S-like declarative language for describing graphical models
- Stochastic relations
- Deterministic relations

```
Y ~ dnorm(mu, tau)
tau ~ dgamma(0.01, 0.01)
mu <- alpha + beta * x
```



Summary

1. Context
2. BiiPS software
3. Example in financial econometry

BiiPS software

- Core developped in C++ (>30K lines)
- BUGS language compiler adapted from JAGS © M. Plummer
- Multi-platform: Linux, Windows, Mac OSX
- Open-source GPL license
- RBiips interface for R
- MatBiips interface for Matlab (ongoing development)

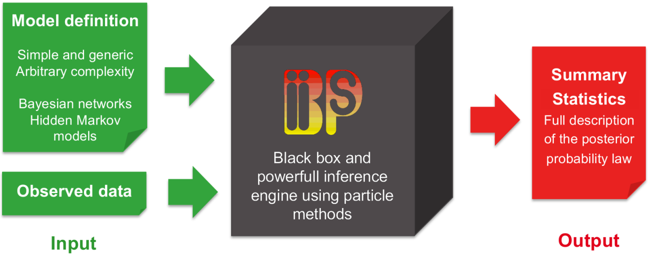


Figure : *BiiPS*: input/output diagram

How to drive vanilla SMC on a graphical model?

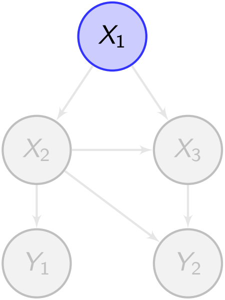


Figure : Directed acyclic graph

- Sample $x_1^{(i)} \sim p(x_1)$, and set weights $w_1^{(i)} = 1/N$
- Sample $x_2^{(i)} \sim p(x_2|x_1^{(i)})$
- Set weights $w_2^{(i)} = w_1^{(i)} p(y_1|x_2^{(i)})$ and resample $\{x_{1:2}^{(i)}, w_2^{(i)}\}$
- Sample $x_3^{(i)} \sim p(x_3|x_{1:2}^{(i)})$
- Set weights $w_3^{(i)} = w_2^{(i)} p(y_2|x_{2:3}^{(i)})$

How to drive vanilla SMC on a graphical model?

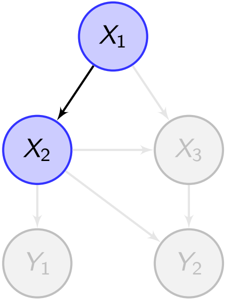


Figure : Directed acyclic graph

- Sample $x_1^{(i)} \sim p(x_1)$, and set weights $w_1^{(i)} = 1/N$
- Sample $x_2^{(i)} \sim p(x_2|x_1^{(i)})$
- Set weights $w_2^{(i)} = w_1^{(i)} p(y_1|x_2^{(i)})$ and resample $\{x_{1:2}^{(i)}, w_2^{(i)}\}$
- Sample $x_3^{(i)} \sim p(x_3|x_{1:2}^{(i)})$
- Set weights $w_3^{(i)} = w_2^{(i)} p(y_2|x_{2:3}^{(i)})$

How to drive vanilla SMC on a graphical model?

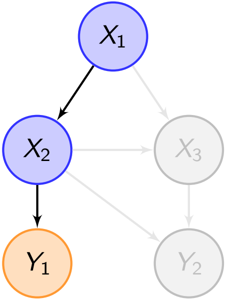


Figure : Directed acyclic graph

- Sample $x_1^{(i)} \sim p(x_1)$, and set weights $w_1^{(i)} = 1/N$
- Sample $x_2^{(i)} \sim p(x_2|x_1^{(i)})$
- Set weights $w_2^{(i)} = w_1^{(i)} p(y_1|x_2^{(i)})$ and resample $\{x_{1:2}^{(i)}, w_2^{(i)}\}$
- Sample $x_3^{(i)} \sim p(x_3|x_{1:2}^{(i)})$
- Set weights $w_3^{(i)} = w_2^{(i)} p(y_2|x_{2:3}^{(i)})$

How to drive vanilla SMC on a graphical model?

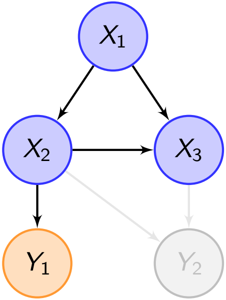


Figure : Directed acyclic graph

- Sample $x_1^{(i)} \sim p(x_1)$, and set weights $w_1^{(i)} = 1/N$
- Sample $x_2^{(i)} \sim p(x_2|x_1^{(i)})$
- Set weights $w_2^{(i)} = w_1^{(i)} p(y_1|x_2^{(i)})$ and resample $\{x_{1:2}^{(i)}, w_2^{(i)}\}$
- Sample $x_3^{(i)} \sim p(x_3|x_{1:2}^{(i)})$
- Set weights $w_3^{(i)} = w_2^{(i)} p(y_2|x_{2:3}^{(i)})$

How to drive vanilla SMC on a graphical model?

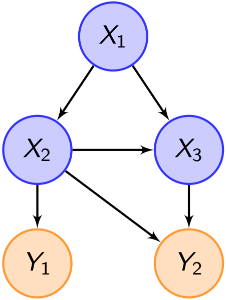


Figure : Directed acyclic graph

- Sample $x_1^{(i)} \sim p(x_1)$, and set weights $w_1^{(i)} = 1/N$
- Sample $x_2^{(i)} \sim p(x_2|x_1^{(i)})$
- Set weights $w_2^{(i)} = w_1^{(i)} p(y_1|x_2^{(i)})$ and resample $\{x_{1:2}^{(i)}, w_2^{(i)}\}$
- Sample $x_3^{(i)} \sim p(x_3|x_{1:2}^{(i)})$
- Set weights $w_3^{(i)} = w_2^{(i)} p(y_2|x_{2:3}^{(i)})$

How to drive vanilla SMC on a graphical model?

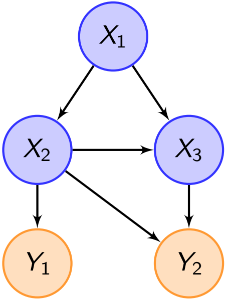


Figure : Directed acyclic graph

More generally:

- Sample nodes in a topological order
- One iteration of SMC corresponds to sampling one unobserved parameter: $p(X_k|\text{parents}(X_k))$
- Weight particles with likelihood associated to their observed children $p(Y_k|\text{parents}(Y_k))$

Choice of the importance distribution

- Sampling from the prior $p(X_k|\text{parents}(X_k))$ may lead to bad resampling and degeneracy problems
- It is better to sample from $p(X_k|\text{parents}(X_k), \text{children}(X_k))$ or any approximation

Prior to running the SMC, *BiiPS* assigns an appropriate importance sampling method (node sampler) to each unobserved parameter

1. Finite discrete sampler
2. Conjugate sampler (only normal prior implemented yet)
3. Prior sampler (default)

When does it work?

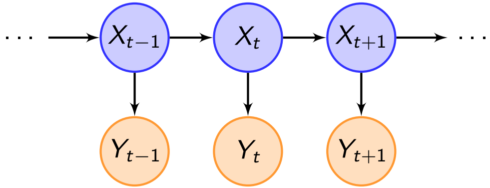


Figure : State-space model / HMM

- Ok with state-space models (HMM), switching state-space models, etc.
- More generally: ok when the unkown parameters are controlled by a dynamic system
- But this method will not suit all graphical models

When does it work?

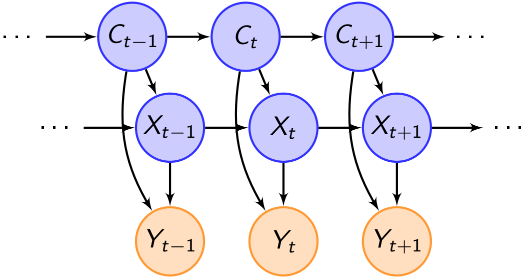
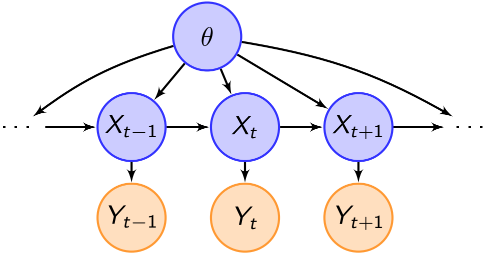


Figure : Switching state-space model

- Ok with state-space models (HMM), switching state-space models, etc.
- More generally: ok when the unkown parameters are controlled by a dynamic system
- But this method will not suit all graphical models

How to deal with fixed parameters?



Particle Marginal Metropolis-Hastings [?]

MCMC algorithm using SMC at each iteration.
At iteration k:

- Propose a θ^*
- Run an SMC algorithm conditionally on θ^*
- Accept or reject θ^* with acceptance rate depending on the estimate of the conditional marginal likelihood $\hat{p}_{\theta^*}(Y_{1:T})$

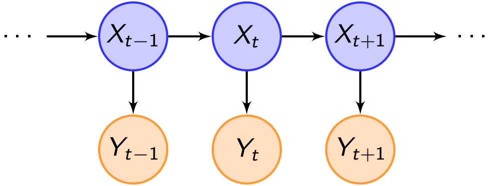
Summary

1. Context
2. BiiPS software
3. Example in financial econometry

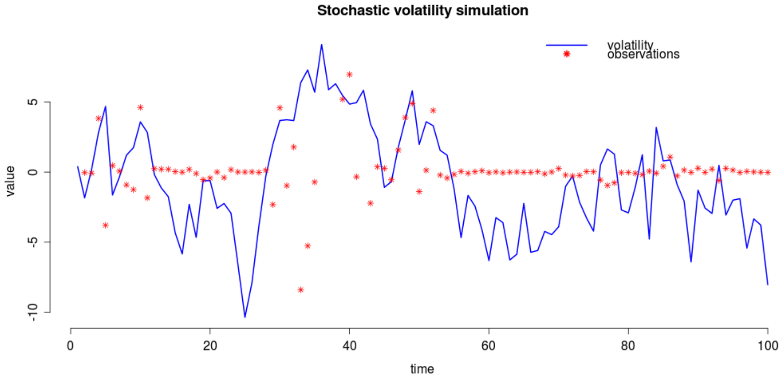
Estimation of the stochastic volatility

Consider infering the underlying volatility $X_{1:T}$ from observed incremental price or rate $Y_{1:T}$

$$X_1 \sim \mathcal{N}(0, \sigma^2)$$
$$X_t | X_{t-1} \sim \mathcal{N}(\alpha x_{t-1}, \frac{\sigma^2}{1-\alpha^2}) \quad t > 1$$
$$Y_t | X_t \sim \mathcal{N}(0, \beta^2 \exp(x_t)) \quad t > 1$$



Estimation of the stochastic volatility



Estimation of the stochastic volatility

BUGS language "volatility.bug"

```
model
{
   x[1] ~ dnorm(0, 1 / sigma^2)
   prec.x <- (1-alpha^2) / sigma^2
   for (t in 2:t.max)
   {
      x[t] ~ dnorm(alpha * x[t-1], prec.x)
      prec.y[t] <- 1 / (beta^2 * exp(x[t]))
      y[t] ~ dnorm(0, prec.y[t])
   }
}
```

Estimation of the stochastic volatility

```
# Load RBiips
library(RBiips)

# Define data
data <- list(t.max=100, sigma=1.0,
             alpha=0.91, beta=0.5,
             y=y)

# Compile the model and load the data
model <- biips.model("volatility.bug", data)

# Run SMC algorithm
out.smc <- smc.samples(model, "x", n.part=1000)
```

Estimation of the stochastic volatility

```
# Summary statistics
x.summ <- summary(out.smc$x, fun=c("mean","quantiles"),
                  probs=c(.05,.95))
plot(x.summ)
```

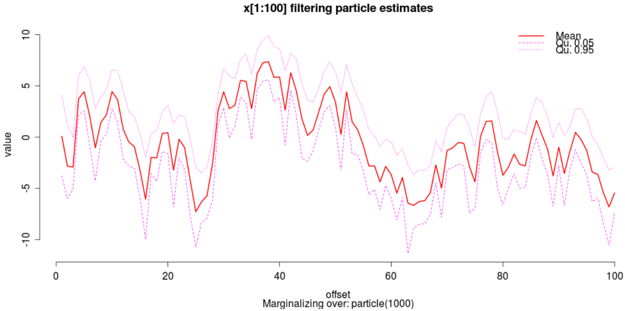


Figure : Summary statistics

Estimation of the stochastic volatility

```
# Kernel density estimates
plot(density(out.smc$x, adjust=2))
```

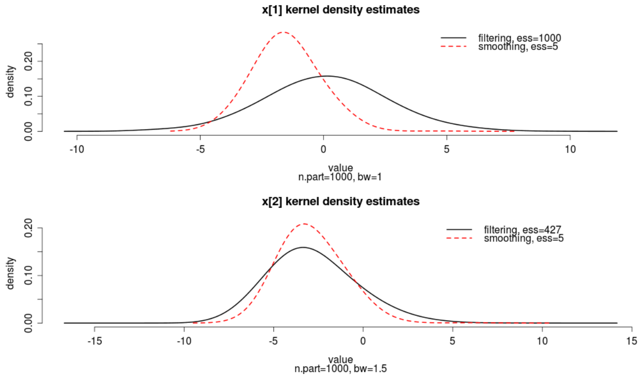
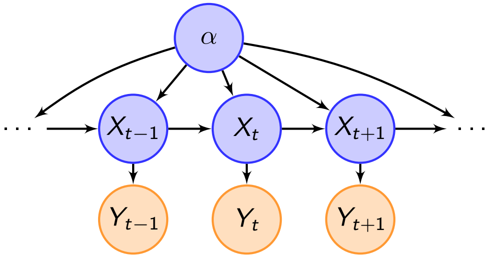


Figure : Kernel density estimates

Estimation of the fixed parameter α

BUGS language "volatility_param.bug"

```
model {
   x[1] ~ dnorm(0, 1 / sigma^2)
   prec.x <- (1-alpha^2) / sigma^2
   for (t in 2:t.max) {
      f[t] <- alpha * x[t-1]
      x[t] ~ dnorm(f[t], prec.x)
      prec.y[t] <- 1 / (beta^2 * exp(x[t]))
      y[t] ~ dnorm(0, prec.y[t])   }
   alpha ~ dunif(0, 0.99)   }
```



Estimation of the fixed parameter α

```
# Define data and compile the model
data <- list(t.max=100, sigma=1.0,
             beta=0.5, y=y)
model <- biips.model("volatility_param.bug", data)

# Burn in PMMH algorithm
update.pmmh(model, "alpha", n.iter=1000, n.part=100)

# Generate PMMH samples
out.pmmh <- pmmh.samples(model, "alpha", n.iter=10000,
                         n.part=100)

# Summary statistics
alpha.mean <- mean(out.pmmh$alpha)
alpha.var <- var(out.pmmh$alpha)
```

Estimation of the fixed parameter α

```
# PMMH trace plot and histogram
plot(out.pmmh$alpha)
hist(out.pmmh$alpha)
```

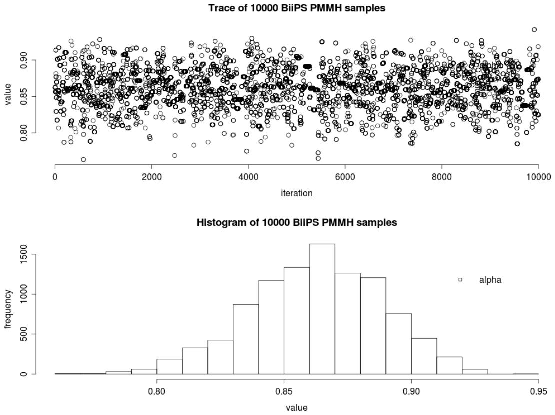


Figure : α PMMH samples: trace plot and histogram

THANK YOU

http://alea.bordeaux.inria.fr/biips

adrien.todeschini@inria.fr

Centre de Bordeaux

200 avenue de la Vieille Tour
33405 Talence Cedex, France

www.inria.com