

Markov chain Monte Carlo methods for the Bayesian analysis of stochastic process models

Dr David A. Stephens

Department of Mathematics & Statistics

McGill University

david.stephens@mcgill.ca

<https://www.math.mcgill.ca/dstephens/SSC2025/>



1. Stochastic process models & applications
2. Monte Carlo methods
 - Markov chain Monte Carlo (MCMC)
 - Sequential Monte Carlo (SMC)
 - Particle MCMC
3. Gaussian Models
4. Hidden Markov Models
5. Stochastic Epidemic Models
6. Volatility Models & Models driven by Lévy processes

Part 1

Introduction

Models based on stochastic processes occur in many application areas

- longitudinal
 - ▶ time series modelling
 - ▶ financial applications
 - ▶ medical monitoring
- spatial
 - ▶ climate & weather
 - ▶ spatial epidemiology
 - ▶ point incidence

Today: only time to focus on 1-d problems !

General notation:

- Y is an observed quantity
- X is an auxiliary quantity
- $\theta \in \mathbb{R}^d$ is a parameter
- Stochastic processes will be denoted $X(t), Y(t), Z(t)$ etc.
- Discrete time observations will be denoted X_t, Y_t, Z_t etc.
- t will be a time index
- i will be a data index

Simple stochastic model: based on the SDE

$$dX(t) = b(X(t), t)dt + \sigma(X(t), t)dW(t)$$

where $W(t)$ is a standard *Brownian motion*

- (I) $W(0) = 0$
- (II) $W(t)$ is stochastically continuous
- (III) $W(t)$ has independent Gaussian increments: for $s > t$,

$$W(s) - W(t) \sim \text{Normal}(0, (s - t))$$

with

$$W(s_1) - W(t_1), W(s_2) - W(t_2)$$

independent for non-overlapping intervals $(t_1, s_1), (t_2, s_2)$.

Models based on stochastic processes

- $b(X(t), t)dt$ is the drift
- $\sigma(X(t), t)dW(t)$ is the diffusion term

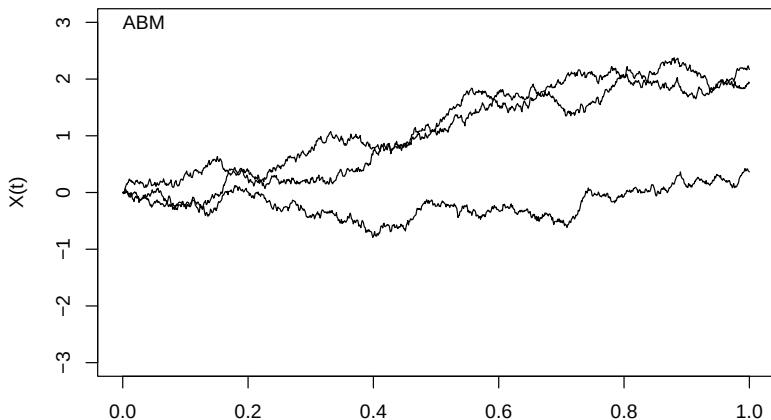
Captures many forms of stochastic behaviour.

Integral form:

$$X(t) = X(0) + \int_0^t b(X(s), s) ds + \int_0^t \sigma(X(s), s) dW(s)$$

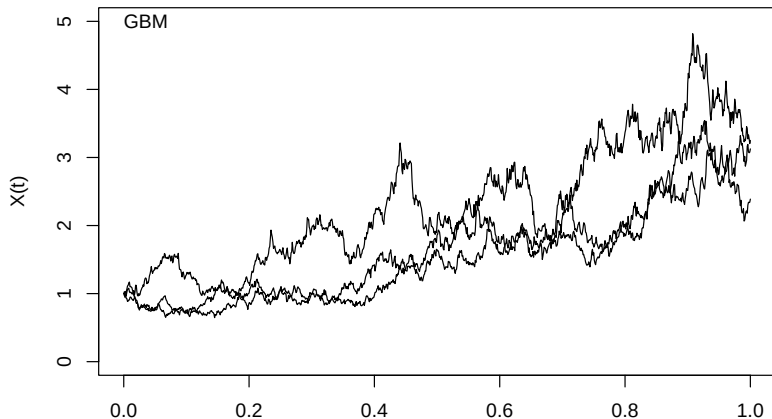
Models based on stochastic processes

$$dX(t) = \theta dt + \sigma dW(t)$$



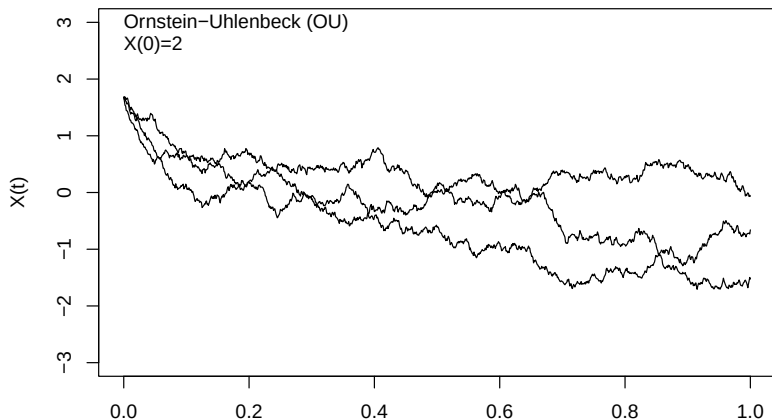
Models based on stochastic processes

$$dX(t) = \theta X(t)dt + \sigma X(t)dW(t)$$



Models based on stochastic processes

$$dX(t) = -\mu X(t)dt + \sigma dW(t)$$



Inference:

(I) *Directly observed case*: We may observe data

$$X_{t_1}, X_{t_2}, \dots, X_{t_n}$$

and wish to infer the parameters of the process

(II) *Indirectly observed case*: We may observe data

$$Y_{t_1}, Y_{t_2}, \dots, Y_{t_n}$$

and wish to infer the parameters of the process *and* the underlying process itself.

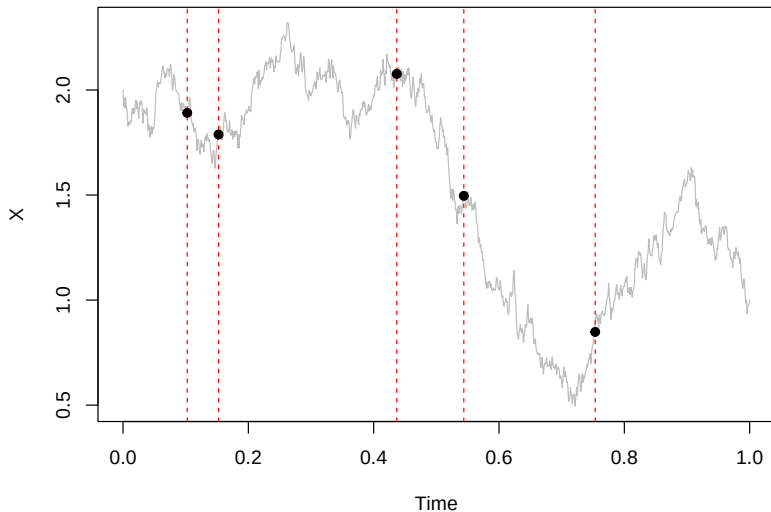
Two cases: observation times

$$t_1, t_2, \dots, t_n, \dots$$

are

- *equally-spaced* (e.g. 0, 1, 2, ...)
- *unequally-spaced* (e.g. 0, 1.2, 6.2, 6.5, 11.4, ...)

Models based on stochastic processes



Most tractable methods depend on *discretization* strategies: for example

$$X_{t+\delta t} - X_t = \mu(X_t)\delta t + \sigma(X_t)W_t$$

where $W_t \sim \text{Normal}(0, (\delta t)^2)$, so that

$$X_{t+\delta t}|X_t \sim \text{Normal}(X_t + \mu(X_t)\delta t, (\sigma(X_t)\delta t)^2)$$

This is the strategy we shall mostly adopt.

Note: for diffusions and some other models, exact methods do exist.

Critical in this discretization is the availability of the *transition density*

$$p_{X_t|X_s}(x_t|x_s) \quad t > s$$

as this allows e.g. simulation and likelihood evaluation.

The main statistical challenge in general is the lack of an explicit form for this density. This arises when the *driving* process $W(t)$ is replaced by a more general process.

Example: Model driven by a Lévy process

A model used in financial applications is the OU process driven by Lévy process $\{Z(t)\}$.

$$dX(t) = -\lambda X(t)dt + dZ(\lambda t)$$

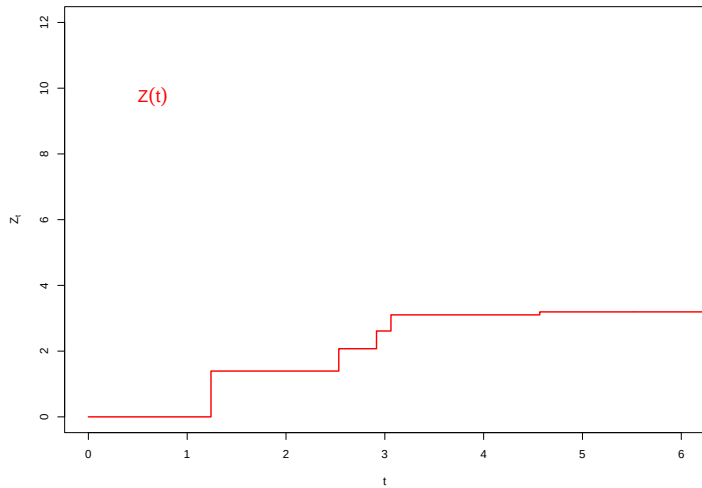
or equivalently

$$X(t) = e^{-\lambda t}X(0) + e^{-\lambda t} \int_0^t e^{\lambda s} dZ(\lambda s)$$

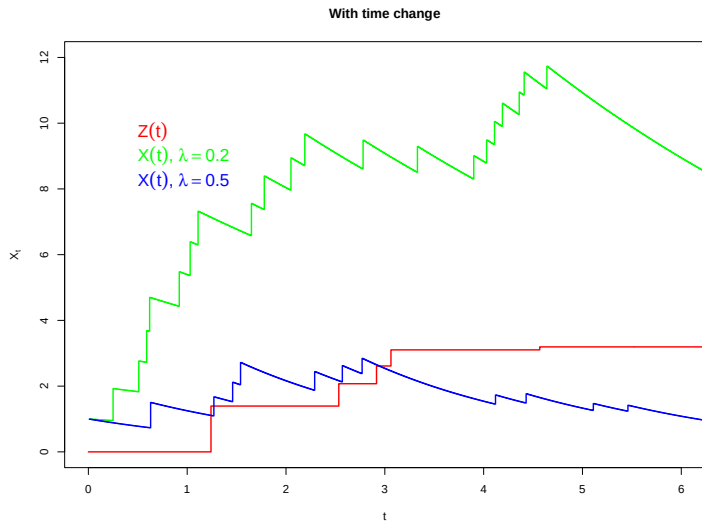
$\lambda > 0$ induces a *time change*.

For example, $\{Z(t)\}$ may be a *pure jumps* process.

Models based on stochastic processes



Models based on stochastic processes



Example: Model driven by a Lévy process

We may attempt to address this via the stochastic integral

$$\int_0^t e^{\lambda s} dZ(\lambda s) \stackrel{d}{=} \sum_{j=1}^{\infty} e^{\lambda R_j} Z_j$$

for appropriately generated (R_j, Z_j) .

In this model we can simulate from the transition density, but evaluation is more complex.

1.2. Markov chain models

Assume $\{X(t)\}$ takes values on a countable set $\{1, 2, \dots, K, \dots\}$ where the dynamics are specified via the conditional probabilities

$$\Pr[X_t = x_t | X_s = x_s]$$

with observations being made in

- discrete time: $t = 1, 2, \dots$
- continuous time: $t = t_1, t_2, \dots$

This model may be used for

- the observed data
- as a *latent* or *hidden* process
- transition times between states may be observed or unobserved.

1.2. Markov chain models

The model is characterized by

- initialization quantities $X(0)$
- dynamics parameters
 - ▶ transition matrix, $\mathbf{P}$
 - ▶ generator matrix, $\mathbf{Q}$

which are the focus of inferential interest.

1.3. Bayesian Inference

We assume the availability of observed data

$$y_1, y_2, \dots, y_n$$

presumed generated from a stochastic data generating model parameterized by finite dimensional parameter θ .

We begin by assuming the corresponding random variables

$$Y_1, Y_2, \dots, Y_n$$

are conditionally independent given θ .

1.3. Bayesian Inference

IID Case: We represent the data generating model for the observables as

$$f_{Y_1, \dots, Y_n}(y_1, \dots, y_n) = \int \prod_{i=1}^n f(y_i; \theta) \pi_0(d\theta)$$

where

$$f_{Y_1, \dots, Y_n}(y_1, \dots, y_n)$$

is the *joint pdf* for $Y_1, \dots, Y_n$. The term

$$\prod_{i=1}^n f_{Y_i}(y_i; \theta)$$

represents the *likelihood function*, $\mathcal{L}_n(\theta)$.

1.3. Bayesian Inference

The *posterior distribution* is

$$\pi_n(\theta) = \frac{\prod_{i=1}^n \mathcal{L}_n(\theta) \pi_0(\theta)}{\int \prod_{i=1}^n \mathcal{L}_n(t) \pi_0(t) dt}$$

where

$$\pi_n(\theta)$$

is an *updated* version of *prior distribution*

$$\pi_0(\theta)$$

in light of observing $y_1, \dots, y_n$.

1.3. Bayesian Inference

In Bayesian inference, the posterior forms the basis of inference

- estimation
- uncertainty representation
- prediction

which can all be achieved in a fully probabilistic and coherent fashion.

Our task: compute $\pi_n(\theta)$.

Note

The formulation extends beyond the conditionally independent case e.g.

- dependent data
- hierarchical models

but the principles that lead to the computation of the posterior distribution are the same.

1.4. Augmenting the parameter space

In some models, the likelihood is difficult to compute directly, but computation can be simplified by introducing *auxiliary* variables. Suppose

$$f(y; \theta) = \int f(y, x; \theta) dx$$

say, where $f(y, x; \theta)$ is more tractable than $f(y; \theta)$. We can treat x as a further unknown quantity, and compute the joint posterior.

The quantity X could be

- an additional parameter (e.g. Student-t likelihood)
- missing data (e.g. mixture model)

1.4. Augmenting the parameter space

Let $\mathbf{X}$ be the vector of auxiliary variables. We then compute the joint posterior

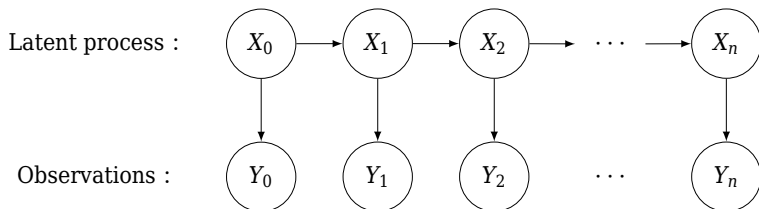
$$\pi_n(\theta, \mathbf{X})$$

which has the target posterior

$$\pi_n(\theta)$$

as its marginal distribution if we integrate out over $\mathbf{X}$.

Dependence via conditional independence



- $Y_0, Y_1, \dots, Y_n$ *conditionally independent* given $X_0, X_1, \dots, X_n$;
- marginally $Y_0, Y_1, \dots, Y_n$ are *dependent*.
- θ characterizes the properties of the $\{X(t)\}$ process and the observation process.

This is the form of a *state space* and *hidden Markov models*.

Example: Health trajectories

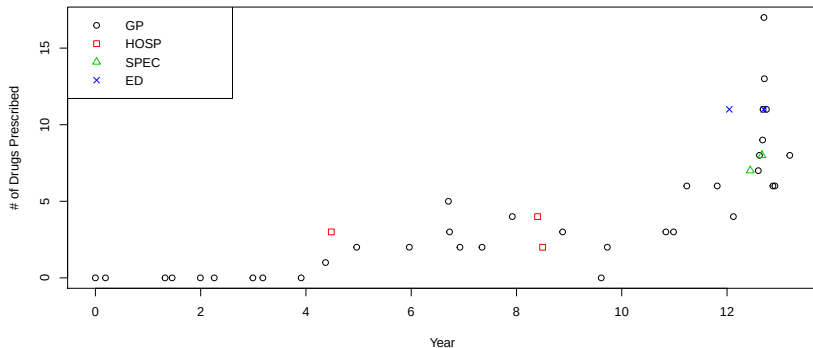
In health surveillance, we may observe health system utilizations

- primary care visits
- prescription drug usage
- emergency care visits

and attempt to infer the underlying health status of the individual.

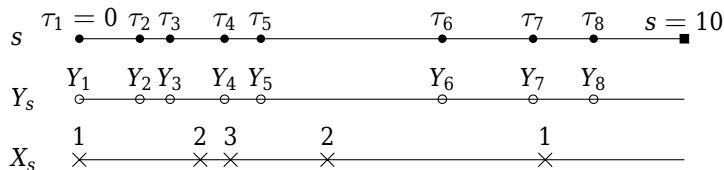
We may model the health status as a hidden *continuous time Markov chain* (CTMC), with conditionally independent utilization observations.

COPD cohort: one trajectory



Luo et. al. (2021)

Dependence via conditional independence

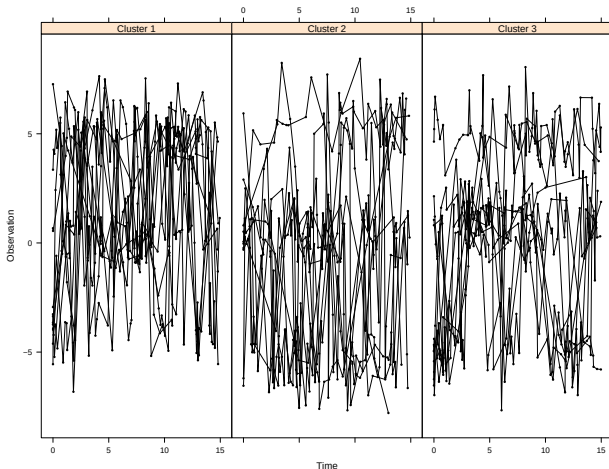


$\{X(t)\}$ is the latent health states, and follows a CTMC.

Notice that the state changes in $\{X(t)\}$ do not occur at the same times as the observations are made.

Dependence via conditional independence

COPD cohort: cluster structure



Luo et. al. (2023)

Part 2

Monte Carlo methods

For probability density $f(x)$, we first consider approximating

$$\mathbb{E}_f[g(X)] = \int g(x)f(x) dx < \infty$$

by

$$\hat{I}_N(g) = \frac{1}{N} \sum_{i=1}^N g(x_i)$$

where $x_1, \dots, x_N \sim f$ are a sample generated from f .

This is the basic *Monte Carlo* calculation.

Typically we have that

$$\widehat{I}_N(g) \xrightarrow[p]{\text{a.s.}} \mathbb{E}[g(X)] = \mu(g)$$

say, *and* a central limit theorem applies

$$\sqrt{N}(\widehat{I}_N(g) - \mu(g)) \xrightarrow{d} \text{Normal}(0, V(g))$$

where

$$V(g) = \text{Var}[g(X)] = \int (g(x) - \mu(g))^2 f(x) dx$$

Q. Does this ever go wrong ?

A. It can, but with care for standard statistical problems, we can ensure validity.

Q. Is it the best we can do ?

A. Maybe sampling from another distribution rather than f can improve things ?

Q. Is sampling from f easy?

A. No, in interesting problems, usually not.

If f_0 is a pdf with support including the support of f , then

$$\int g(x)f(x) dx = \int g(x)f(x)\frac{f_0(x)}{f_0(x)} dx = \int \frac{g(x)f(x)}{f_0(x)}f_0(x) dx.$$

and so

$$\mathbb{E}_f[g(X)] = \mathbb{E}_{f_0}\left[\frac{g(X)f(X)}{f_0(X)}\right].$$

Hence if $X_1, \dots, X_N \sim f_0(\cdot)$ an estimator of the expectation is

$$\hat{I}_N^{(f_0)}(g) = \frac{1}{N} \sum_{i=1}^N \frac{g(X_i)f(X_i)}{f_0(X_i)}$$

$\hat{I}_N^{(f_0)}$ is termed the *importance sampling* estimator, and f_0 is termed the *importance sampling density*.

The optimal choice of f_0 can be shown to be

$$f_0(x) \propto |g(x)|f(x).$$

i.e. need f_0 to have high density where the integrand is large

Note: If

$$\frac{f(x)}{f_0(x)}$$

is *unbounded* on the support of f , the variance of $\widehat{I}_N^{(f_0)}(g)$ is *not finite*. Therefore, for f with unbounded support, we must ensure that this ratio stays bounded particularly in the *tails*.

All of the previous results carry across to the case where the target integral is an integral in dimension higher than one.

- $g(\mathbf{x})$ is a scalar function of vector argument $\mathbf{x}$,
- $f(\mathbf{x})$ and $f_0(\mathbf{x})$ are multivariate densities.

In higher dimensions, in general, many more random samples are needed to obtain sufficient accuracy than in the univariate case.

Sampling Importance Resampling (SIR)

1. Generate variates $x_1, \dots, x_N$ from f_0 .
2. Compute normalized weights $w_1, \dots, w_N$ given by

$$w_i = \frac{f(x_i)/f_0(x_i)}{\sum_{j=1}^N (f(x_j)/f_0(x_j))} \quad i = 1, \dots, N.$$

3. Resample y_i from $\{x_1, \dots, x_N\}$ with masses $\{w_1, \dots, w_N\}$.

Part 3

Markov chain Monte Carlo

Generating from $f(x)$ is not straightforward when

- x is high-dimensional (but not Gaussian),
- $f(x)$ is of non-standard form.

These difficulties occur often (but not exclusively) in Bayesian inference.

We seek a way of simulating from $f(x)$ in this situation.

Markov chain Monte Carlo (MCMC) operates by

- constructing a *Markov chain* that has the posterior distribution as its stationary/equilibrium distribution
- running the chain forward and collecting the sampled states.

Most typically MCMC simulates a *discrete time* Markov chain, sampling

$$\theta^{(1)}, \theta^{(2)}, \dots, \theta^{(N)}$$

over some finite number of iterations, N .

- If the posterior distribution is *discrete* then we may use a conventional Markov chain construction via a transition matrix P .
- More typically, $\pi_n(\theta)$ will have *continuous* components so a slightly more general construction is warranted.

In the *continuous state* case, we must specify a *transition kernel*

$$P(x, B) = \int_B P(x, z) dz$$

$P(x, B)$ determines the probability of making the transition from current value x into the set $B \subset \mathbb{X}$ in any given step.

- Transitions are implemented using a *transition density*

$$P(x, z) \equiv P(x \rightarrow z)$$

which specifies a conditional probability density in z , given the current value x , for $x, z \in \mathbb{X}$.

- The stationary distribution π for the continuous case must satisfy

$$\pi(x) = \int P(z, x) \pi(z) dz$$

- A reversible chain must satisfy detailed balance

$$\pi(x)P(x, z) = \pi(z)P(z, x)$$

for all x and z . Given P , we can in theory solve for π .

In the context of sampling from a *target* probability distributions, we wish to *specify* π , and *then find* a P such that its equilibrium distribution is π .

The Metropolis-Hastings Algorithm

Define *proposal density* q such that

$$q(z, x) = q(z \rightarrow x)$$

facilitates sufficient movement across the whole of the parameter space.

Then, for $z \neq x$, define

$$P(x, z) = q(x, z) \alpha(x, z)$$

where

$$\alpha(x, z) = \min \left\{ 1, \frac{\pi(z)}{\pi(x)} \frac{q(z, x)}{q(x, z)} \right\}$$

defines an *acceptance probability* for the move from x to z .

The Metropolis-Hastings Algorithm

Thus, to generate the Markov chain, if $X_n = x$, we

1. Generate z from conditional density $q(x, \cdot)$ given x ;
2. Compute $\alpha(x, z)$;
3. Generate u from *Uniform* $(0, 1)$
 - if $u \leq \alpha(x, z)$, *accept* the move to z and set $X_{n+1} = z$
 - if $u > \alpha(x, z)$, *reject* the move to z and set $X_{n+1} = x$
4. Return to 1. to generate X_{n+2}

and so on.

This is the *Metropolis-Hastings (MH)* algorithm.

The Metropolis-Hastings Algorithm

- *Metropolis Algorithm*: Choose $q(x, z) = q(z, x)$
 - ▶ e.g. $Z|X_n = x \sim \text{Normal}(x, \sigma_q^2)$, then

$$\alpha(x, z) = \min \left\{ 1, \frac{\pi(z)}{\pi(x)} \right\}$$

and the move to z is accepted *with certainty* if $\pi(z) \geq \pi(x)$.

- *Independence MH*: Choose $q(x, z) = q(z)$
 - ▶ i.e. *independent* of the current value of the chain. Then

$$\alpha(x, z) = \min \left\{ 1, \frac{\pi(z) q(x)}{\pi(x) q(z)} \right\}$$

We also may *mix* several different proposal densities.

In practice seek to choose the proposal mechanism so that

- an *irreducibility* condition is met
- the chain explores the parameter space rapidly (*fast mixing*)
- the *autocorrelation* of sampled values is as small as possible
- we monitor the effectiveness of the MCMC implementation:
 - ▶ trace plots, scatter plots
 - ▶ convergence diagnostics
 - ▶ *effective sample size*

$$N_{\text{eff}} = \frac{N}{1 + 2 \sum_{k=1}^{\infty} \rho(k)}$$

where $\rho(k)$ is the true lag- k acf for the Markov chain.

3.5. The Gibbs sampler

Suppose that π is a probability density in K dimensions, and let the variables be denoted $(X_1, \dots, X_K)$. Define the conditional density $\pi_k(.|.)$ for

$$X_k | X_1, \dots, X_{k-1}, X_{k+1}, \dots, X_K \equiv X_k | X_{(-k)}$$

by

$$\pi_k(x_k | x_{(-k)}) = \frac{\pi(x_1, \dots, x_K)}{\pi(x_1, \dots, x_{k-1}, x_{k+1}, \dots, x_K)} \propto \pi(x_1, \dots, x_K)$$

where the denominator is the marginal distribution of $X_{(-k)}$.

3.5. The Gibbs sampler

The *Gibbs sampler* is implemented as follows:

1. Set starting values for the K variables $(x_{1,0}, \dots, x_{K,0})$.
2. Sample the conditional distributions with updating:
 - (a) sample $x_{1,1}$ from $\pi_1(x_1 | x_{2,0}, x_{3,0}, \dots, x_{K,0})$
 - (b) sample $x_{2,1}$ from $\pi_2(x_2 | x_{1,1}, x_{3,0}, \dots, x_{K,0})$
 - (c) sample $x_{3,1}$ from $\pi_3(x_3 | x_{1,1}, x_{2,1}, \dots, x_{K,0})$
 - $\vdots$
 - (K) sample $x_{K,1}$ from $\pi_K(x_K | x_{1,1}, x_{2,1}, \dots, x_{K-1,1})$

This completes *one* step of the Gibbs sampler.

3. Return to 2 (a), and repeat to obtain, at step m , the sampled variates $(x_{1,m}, x_{2,m}, \dots, x_{K,m})$

The Gibbs sampler is a special case of the MH algorithm.

3.5. The Gibbs sampler

The updates can be achieved using direct sampling from the conditional distribution or individual MH steps, with acceptance probabilities

$$\alpha_k(x, z) = \min \left\{ 1, \frac{\pi_k(z|x_{(-k)}) q_k(z, x)}{\pi_k(x|x_{(-k)}) q_k(x, z)} \right\}$$

for $k = 1, \dots, K$. In 2., the steps can also be completed in *random order*

Note: can implement the Gibbs sampler on *blocks* of variables.

Note

- (a) performance is hindered when parameters are *highly correlated* in the posterior;
- (b) choosing the *starting values* appropriately is important;
- (c) if we use augmented models (e.g. auxiliary parameters or data), then this increase in the number of parameters can increase the computation time and memory requirements.

The usual strategies for models parameterized using a latent structure is the Gibbs sampler applied to the conditionals

- $\pi_n(\theta|\mathbf{x}; \mathbf{y})$: inference from the complete data model;
- $\pi_n(\mathbf{x}|\theta; \mathbf{y})$: inference for the latent states.

For example, $\pi_n(\mathbf{x}|\theta; \mathbf{y})$ may be sampled by *filtering* approaches

$$\pi_n(\mathbf{x}|\theta; \mathbf{y}) = \pi_n(x_1|\theta; \mathbf{y}) \prod_{t=1}^{n-1} \pi_n(x_{t+1}|x_{1:t}, \theta; \mathbf{y}).$$

If it can be implemented, this *forward sampling* is optimal.

It is also possible to use Gibbs sampling on the conditionals

$$\pi_n(x_t | x_{(-t)}, \theta; \mathbf{y})$$

which simplifies under a Markov assumption to

$$\pi_n(x_t | x_{t-1}, x_{t+1}, \theta; \mathbf{y}).$$

However, it may be that the components of $\mathbf{x}$ are *highly correlated* locally.

This can be a problem for stochastic process models: for example,

$$\pi_n(x_t|x_{t-1}, x_{t+1}, \theta; \mathbf{y}) \propto p(x_t|x_{t-1}, \theta; \mathbf{y})p(x_{t+1}|x_t, \theta; \mathbf{y})\pi_0(\theta)$$

and it will likely be that the transition density components

- $p(x_t|x_{t-1}, \theta; \mathbf{y})$
- $p(x_{t+1}|x_t, \theta; \mathbf{y})$

induce high dependence.

Part 4

Gaussian models

The simplest, but most common, case arises when the transition density is a *Normal* distribution, say

$$p(x_t|x_s; \theta) \equiv \text{Normal}(\mu(s, x_s, \theta), \sigma^2(s, x_s, \theta))$$

which generally means that the corresponding likelihood is available.

However

- non-linear dependence on θ
- number of observations
- latent structure

all make inference more complicated.

4.1. Linear State Space Models

The principle of a state space model is that an observed series is generated as the *marginal* process derived from a multivariate time series model, structured so that the model is specified via two equations

- the *state* equation
- the *observation* equation

We assume initially that the observations are *equally spaced* in time.

4.1. Linear State Space Models

The general (Markov) *state equation* takes the form

$$\mathbf{X}_{t+1} = \mathbf{g}(\mathbf{X}_t; \theta) + \mathbf{W}_t$$

and the general *observation equation* takes the form

$$\mathbf{Y}_t = \mathbf{f}(\mathbf{X}_t; \theta) + \mathbf{V}_t.$$

The process $\{\mathbf{Y}_t\}$ is observed, the process $\{\mathbf{X}_t\}$ is *latent* or unobserved, and one focus of inferential interest.

Let $\mathbf{y}_{1:t} = \{\mathbf{y}_1, \dots, \mathbf{y}_t\}$.

Inference goals for state space models are

- *filtering*: estimate state at time t from observations *up to* time t

$$p(\mathbf{x}_t | \mathbf{y}_{1:t})$$

- *prediction*: estimate state at time t from observations up to time $t - 1$

$$p(\mathbf{x}_t | \mathbf{y}_{1:(t-1)})$$

- *smoothing*: estimate state at time t from *past* and possibly some *future* observations

$$p(\mathbf{x}_t | \mathbf{y}_{1:n})$$

Initially, we assume θ is known.

Key distributions:

- Filtering

$$p(\mathbf{x}_t | \mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{y}_{1:(t-1)})}{p(\mathbf{y}_t | \mathbf{y}_{1:(t-1)})} = \frac{p(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{y}_{1:(t-1)})}{\int p(\mathbf{y}_t | \mathbf{x}) p(\mathbf{x} | \mathbf{y}_{1:(t-1)}) d\mathbf{x}}$$

- Prediction

$$p(\mathbf{x}_t | \mathbf{y}_{1:(t-1)}) = \int p(\mathbf{x}_t | \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} | \mathbf{y}_{1:(t-1)}) d\mathbf{x}_{t-1}$$

These calculations are carried out recursively; both require integration.

A *linear Gaussian* state space model is given by the equations

$$\mathbf{Y}_t = \mathbf{F}_t \mathbf{X}_t + \mathbf{V}_t$$

$$\mathbf{X}_{t+1} = \mathbf{G}_t \mathbf{X}_t + \mathbf{W}_t$$

- $\mathbf{Y}_t$ is $(K \times 1)$
- $\mathbf{X}_t$ is $(L \times 1)$
- $\mathbf{F}_t$ is $(K \times L)$ is non-stochastic
- $\mathbf{G}_t$ is $(L \times L)$ is non-stochastic
- $\{\mathbf{V}_t\}$ and $\{\mathbf{W}_t\}$ are (uncorrelated) Gaussian series
- $\mathbf{X}_1$ has unconditional mean and variance $\mathbf{a}_{1|0}$ and $\mathbf{P}_{1|0}$.

4.2. Kalman Filter

The Kalman filter recursively computes the quantities

- $\mathbf{a}_{t|t} = \mathbb{E}[\mathbf{X}_t | \mathbf{y}_{1:t}]$,
- $\mathbf{a}_{t+1|t} = \mathbb{E}[\mathbf{X}_{t+1} | \mathbf{y}_{1:t}]$,
- $\mathbf{P}_{t|t} = \text{Var}[\mathbf{X}_t | \mathbf{y}_{1:t}]$,
- $\mathbf{P}_{t+1|t} = \text{Var}[\mathbf{X}_{t+1} | \mathbf{y}_{1:t}]$.

These quantities *characterize* the filtering and prediction distributions.

These quantities are obtained by running for $t = 1, \dots, n$ the recursions

$$\mathbf{r}_t = \mathbf{y}_t - \mathbf{F}_t \mathbf{a}_{t|t-1}$$

$$\mathbf{M}_t = \mathbf{F}_t \mathbf{P}_{t|t-1} \mathbf{F}_t^\top + \Sigma_t$$

$$\mathbf{a}_{t|t} = \mathbf{a}_{t|t-1} + \mathbf{P}_{t|t-1} \mathbf{F}_t^\top \mathbf{M}_t^{-1} \mathbf{r}_t$$

$$\mathbf{P}_{t|t} = \mathbf{P}_{t|t-1} - \mathbf{P}_{t|t-1} \mathbf{F}_t^\top \mathbf{M}_t^{-1} \mathbf{F}_t \mathbf{P}_{t|t-1}$$

$$\mathbf{a}_{t+1|t} = \mathbf{G}_t \mathbf{a}_{t|t},$$

$$\mathbf{P}_{t+1|t} = \mathbf{G}_t \mathbf{P}_{t|t} \mathbf{G}_t^\top + \Omega_t$$

The likelihood for data $\mathbf{y} = (\mathbf{y}_1, \dots, \mathbf{y}_n)$ can be constructed as

$$p(\mathbf{y}_1, \dots, \mathbf{y}_n) = p(\mathbf{y}_n | \mathbf{y}_{(n-1)}) \cdots p(\mathbf{y}_2 | \mathbf{y}_1) p(\mathbf{y}_1) = \prod_{t=1}^n p(\mathbf{y}_t | \mathbf{y}_{t-1})$$

We have that

$$p(\mathbf{y}_t | \mathbf{y}_{1:(t-1)}) \equiv \text{Normal}(\mathbf{F}_t \mathbf{a}_{t|t-1}, \mathbf{M}_t).$$

and the log-likelihood function is given by

$$-\frac{1}{2} \sum_{t=1}^n \log(\det \mathbf{M}_t) - \frac{1}{2} \sum_{t=1}^n \mathbf{r}_t^\top \mathbf{M}_t^{-1} \mathbf{r}_t$$

We may compute

$$\mathbf{a}_{t|n} = \mathbb{E}[\mathbf{X}_t | \mathbf{y}_{1:n}]$$

along with its MSE, $\mathbf{P}_{t|n}$ recursively, for all $t = 1, \dots, n - 1$.

In particular, to obtain $\mathbf{a}_{t|n}$ and $\mathbf{P}_{t|n}$, we start with $\mathbf{a}_{n|n}$ and $\mathbf{P}_{n|n}$ and we run backwards for $t = n - 1, \dots, 0$

$$\mathbf{a}_{t|n} = \mathbf{a}_{t|t} + \mathbf{P}_t^* (\mathbf{a}_{t+1|n} - \mathbf{a}_{t+1|t})$$

$$\mathbf{P}_{t|n} = \mathbf{P}_{t|t} + \mathbf{P}_t^* (\mathbf{P}_{t+1|n} - \mathbf{P}_{t+1|t}) \mathbf{P}_t^*$$

$$\mathbf{P}_t^* = \mathbf{P}_{t|t} \mathbf{G}_t^\top \mathbf{P}_{t+1|t}^{-1}.$$

Consider the model specified by the linear system

$$Y_t = X_t + V_t$$

$$X_{t+1} = \mu + \phi(X_t - \mu) + W_t$$

and the densities

$$p_{Y|X}(y_t|x_t) \equiv \text{Normal}(x_t, \sigma_V^2)$$

$$p_X(x_t|x_{t-1}) \equiv \text{Normal}(\mu + \phi(x_{t-1} - \mu), \sigma_W^2)$$

with $p_{X_1}(x) \equiv \text{Normal}(\mu, \sigma_W^2/(1 - \phi^2))$.

This is an AR(1) plus noise model. Here all of the full conditionals are available for sampling, and we can sweep through the sampling efficiently. In R, packages

- `d1m`
- `bssm`
- `rstan`

can be used.

Example

(i) Gibbs sampler: sample

$$\pi(x_t | x_{t-1}, x_{t+1}, y_{1:n})$$

(ii) Backward sweep: sample

$$t = n \quad \pi(x_n | y_{1:n})$$

$$t = n - 1 \quad \pi(x_{n-1} | x_n, \mathbf{y})$$

$$\vdots \quad \vdots$$

$$t = 2 \quad \pi(x_2 | x_{3:n}, \mathbf{y})$$

$$t = 1 \quad \pi(x_1 | x_{2:n}, \mathbf{y})$$

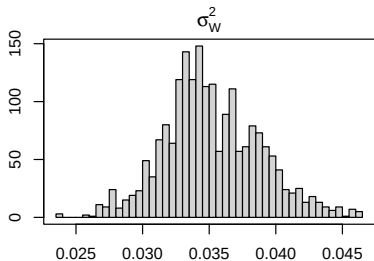
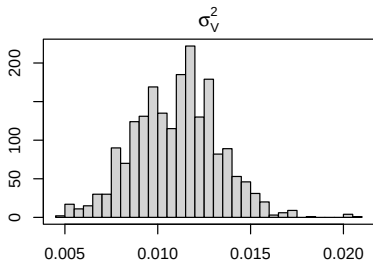
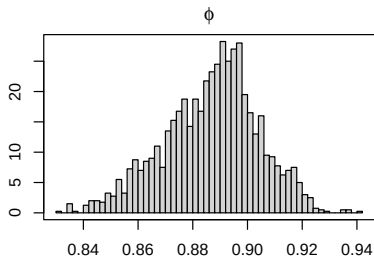
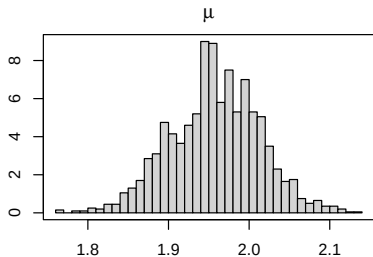
using the univariate Gaussian smoothing distributions.

(iii) Can use `d1mLL` to compute the $\mathcal{L}_n(\theta)$, and then run MH for the posterior $\pi_n(\theta)$

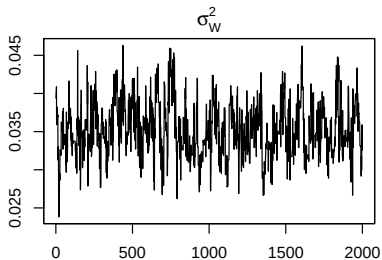
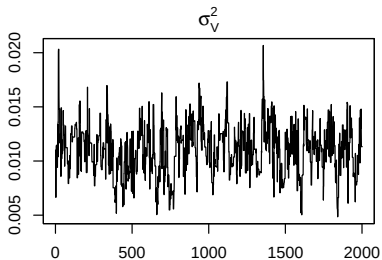
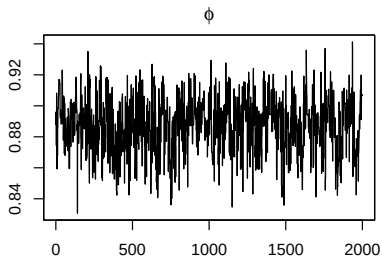
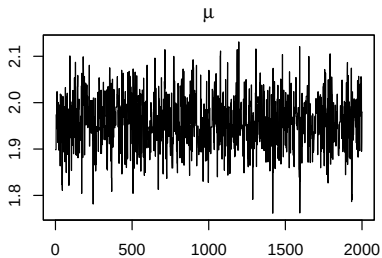
In this case version (iii) is the most efficient

- computation of the filtered likelihood very fast
- integrates out $X_1, \dots, X_n$, so the issue of dependence does not arise
- only a four dimensional posterior

Example: $n = 1000$



Example: $n = 1000$



Example: $n = 1000$

```
library(coda)
effectiveSize(samples)

##      var1      var2      var3      var4
## 847.0544 319.2297 127.5165 126.7795
```

In this model, one can use the

- non-centered parameterization via X_t
- centered parameterization via

$$X_t^* = X_t - \mu$$

which will have different performance in a Gibbs sampler depending on the relative magnitudes of σ_V and σ_W .

4.4. Non-linear or Non-Gaussian models

If the state space model is not linear and Gaussian, then these calculations cannot be completed analytically; specifically

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \quad p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) \quad p(\mathbf{y}_{t+1}|\mathbf{y}_{1:t})$$

will not be available in closed form.

- A pseudo-likelihood can still be constructed in the same way using the minimum mean square linear estimators of the state vector.
- Linear approximation to the non-linearity may be possible

4.4. Non-linear or Non-Gaussian models

For the Bayesian MCMC-based inference the conditional posterior

$$\pi_n(x_t | x_{t-1}, x_{t+1}, \theta; \mathbf{y}) \propto p(x_t | x_{t-1}, \theta; \mathbf{y}) p(x_{t+1} | x_t, \theta; \mathbf{y}) \pi_0(\theta)$$

may be approximated well by an easy to sample distribution.

This facilitates Metropolis-within-Gibbs samplers.

4.5. Particle Filtering

The particle filter uses *importance sampling*, to construct the approximations. Given a non-linear and non-Gaussian state space model, the goal is to compute the filtering and prediction densities

$$p(x_t | y_{1:t}) \quad p(x_{t+1} | y_{1:t})$$

at each time t via a *Sequential Monte Carlo* approximation.

We aim to maintain a sample of *particles*

$$(x_{t,1}, \dots, x_{t,N})$$

for each t that can be used to approximate these distributions.

4.5. Particle Filtering

At each time point t , we perform the

Sequential Importance Sampling

(SIS) procedure:

- Draw random samples from a chosen importance density that can be sampled directly.
- Compute a normalized weight for each particle
- Approximate the target density at time t by the weighted sum of the particles.

Given sufficiently large numbers of particles, this IS characterization converges to the target density.

4.5. Particle Filtering

The general strategy is to build up a series of *particle approximations* to

$$\pi_1(x_1) = p(x_1|y_1)$$

$$\pi_2(x_{1:2}) = p(x_{1:2}|y_{1:2})$$

$$\pi_3(x_{1:3}) = p(x_{1:3}|y_{1:3})$$

$$\vdots$$

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

using features of the state and observation equation.

4.5. Particle Filtering

In the filtering problem, we approximate $p(x_t|y_{1:t})$ by

- particles $(x_{t,1}, \dots, x_{t,N})$, sampled from density f_0 .
- normalized weights $(w_{t,1}^*, \dots, w_{t,N}^*)$

via the discrete distribution $\{(w_{t,i}^*, x_{t,i}), i = 1 \dots, N\}$.

4.5. Particle Filtering

Two key questions are

- What importance density f_0 can or should be chosen?
- How do we compute the weights efficiently (in a recursive fashion) such that for a given weight $w_{t,i}^*$, we just need a new observation y_t to get an approximation to $p(x_t|y_{1:t})$.

A simple but effective choice for f_0 sampling density at time t is the *transition density*: $f_0(x_t|x_{t-1}, y_{1:t}) \equiv p(x_t|x_{t-1})$ as then

$$w_{t,i}^* \propto w_{t-1,i}^* p(y_t|x_{t,i}^*).$$

Recall the state space model: for $t = 1, \dots, n$

$$Y_t = f(X_t; \theta) + V_t$$

$$X_{t+1} = g(X_t; \theta) + W_t$$

where only the series $y_1, \dots, y_n$, is observed. Sequences $\{V_t\}$ and $\{W_t\}$ are independent zero mean random variables.

If $y_{1:t} = (y_1, \dots, y_t)$, $x_{1:t} = (x_1, \dots, x_t)$. we must carry out

- *Filtering*: Compute $p(x_t | y_{1:t})$
- *Prediction*: Compute $p(x_{t+1} | y_{1:t})$
- *Smoothing*: Compute $p(x_t | y_{1:n})$

We consider simulation-based versions of these calculations.

For example, in the *filtering* step, suppose we have a sample

$$z_{t,1}, \dots, z_{t,N}$$

from $p(x_t|y_{1:(t-1)})$ obtained at the previous prediction step. Then we can obtain a *particle approximation* to $p(x_t|y_{1:t})$ as

$$\hat{p}(x_t|y_{1:t}) = \frac{1}{N} \sum_{i=1}^N p(y_t|z_{t,i}) \delta_{z_{t,i}}(x_t)$$

The predictive distribution $p(y_t|y_{1:(t-1)})$ can be approximated by

$$\hat{p}(y_t|y_{1:(t-1)}) = \frac{1}{N} \sum_{i=1}^N p(y_t|z_{t,i})$$

which is useful in approximating the likelihood by

$$\hat{\mathcal{L}}_n(\theta) = \hat{p}(y_1) \prod_{t=2}^n \hat{p}(y_t|y_{1:(t-1)}; \theta)$$

We can produce a sample from the particle approximation

$$\widehat{p}(x_t|y_{1:t})$$

by performing a *weighted resampling* from

$$(z_{t,1}, \dots, z_{t,N})$$

with weights proportional to $p(y_t|z_{t,i})$. Denote this resampled vector

$$v_{t,1}, \dots, v_{t,N}$$

which represent a sample from $p(x_t|y_{1:t})$.

For the *prediction* step, the particle approximation to

$$p(x_{t+1}|y_{1:t})$$

is obtained using a similar Monte Carlo strategy

$$\hat{p}(x_{t+1}|y_{1:t}) = \frac{1}{N} \sum_{i=1}^N p(x_{t+1}|v_{t,i})$$

We propagate the samples $v_{t,1}, \dots, v_{t,N}$ forward by sampling the transition densities for each particle

$$p(x_{t+1}|v_{t,i})$$

This yields the sample $z_{t+1,1}, \dots, z_{t+1,N}$ and the recursion continues.

We use the SIS density

$$p_n(x_{1:n}) = p_1(x_1) \prod_{t=2}^n \tilde{p}_t(x_t | x_{1:(t-1)})$$

with weights given by $w_1(x_1) = \pi_1(x_1)/p_1(x_1)$ and

$$\begin{aligned} w_t(x_{1:t}) &= w_{t-1}(x_{1:(t-1)}) \frac{\tilde{\pi}_t(x_t | x_{1:(t-1)})}{\tilde{p}_t(x_t | x_{1:(t-1)})} \\ &= w_{t-1}(x_{1:(t-1)}) \frac{\pi_t(x_{1:t})}{\pi_{t-1}(x_{1:(t-1)}) \tilde{p}_t(x_t | x_{1:(t-1)})} \end{aligned}$$

$$\begin{aligned} \text{or } w_t(x_{1:t}) &= w_{t-1}(x_{1:(t-1)}) \frac{g_t(x_{1:t})}{g_{t-1}(x_{1:(t-1)}) \tilde{p}_t(x_t | x_{1:(t-1)})} \\ &= w_{t-1}(x_{1:(t-1)}) \alpha_t(x_{1:t}) \end{aligned}$$

if the normalizing constants Z_t for $t = 1, \dots, n$ are unknown.

The Monte Carlo computation is carried out by sampling

$$x_{1,1}, \dots, x_{1,N}$$

from p_1 , then at step $t \geq 2$ sampling for $i = 1, \dots, N$

$$x_{t,1}, \dots, x_{t,N} \quad \text{from} \quad \tilde{p}_t(\cdot | x_{1:(t-1),i})$$

The importance sampling weights are $w_1(x_{1,i})$ and, for $t \geq 2$,

$$w_t(x_{1:t,i}) = w_{t-1}(x_{1:(t-1),i}) \alpha_t(x_{1:t,i})$$

We then have

$$\hat{\pi}_n(\mathbf{x}) = \sum_{i=1}^N W_{n,i} \delta_{x_{1:n,i}}(\mathbf{x})$$

where

$$W_{n,i} = \frac{w_n(x_{1:n,i})}{N\hat{Z}_n} \quad \hat{Z}_n = \frac{1}{N} \sum_{i=1}^N w_n(x_{1:n,i})$$

in the usual way.

We have $X_1 \sim p_{X_1}$ and

$$p_{Y|X}(y_t|x_t) \quad p_X(x_t|x_{t-1})$$

for the transition densities. The posterior of interest is

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

say. We have

$$g_n(x_{1:n}, y_{1:n}) = p_{X_1}(x_1) \left\{ \prod_{t=2}^n p_X(x_t|x_{t-1}) \right\} \left\{ \prod_{t=1}^n p_{Y|X}(y_t|x_t) \right\}$$
$$Z_n(y_{1:n}) = \int g_n(x_{1:n}, y_{1:n}) dx_{1:n}$$

where Z_n may not be available.

For SMC, the optimal choice of IS density $\tilde{p}_t(x_t|x_{1:(t-1)})$ is

$$\tilde{\pi}_t(x_t|x_{1:(t-1)})$$

which in the simple state space model is given by

$$\begin{aligned}\tilde{\pi}_t(x_t|x_{1:(t-1)}) &\equiv \tilde{\pi}_t(x_t|y_{1:t}, x_{1:(t-1)}) = \frac{p_{Y|X}(y_t|x_t)p_X(x_t|x_{t-1})}{p(y_t|x_{t-1})} \\ &= \frac{p_{Y|X}(y_t|x_t)p_X(x_t|x_{t-1})}{\int p_{Y|X}(y_t|x)p_X(x|x_{t-1}) dx}\end{aligned}$$

as, conditional on $x_{1:(t-1)}$, x_t is independent of $y_{1:(t-1)}$.

In most cases the optimal choice is not feasible: we need successive samples from the IS densities

$$\tilde{p}_t(x_t|x_{1:(t-1)}) = \tilde{\pi}_t(x_t|y_{1:t}, x_{1:(t-1)}) = \tilde{\pi}_t(x_t|y_t, x_{(t-1)}).$$

We might choose instead

$$\tilde{p}_t(x_t|x_{1:(t-1)}) = p_X(x_t|x_{t-1})$$

which typically is straightforward to sample from, and yields the recursive weight calculation

$$w_t(x_{1:t}) = w_{t-1}(x_{1:(t-1)})p_{Y|X}(y_t|x_t)$$

However, these weights start to *degenerate* as t increases:

- the weights that define the Monte Carlo estimate $\hat{\pi}_n(\mathbf{x})$ are very small for most i , with only a few large weights.
- the empirical variance of the collection of unnormalized weights $\{w_t(x_{t,i})\}$ for $i = 1, \dots, N$ increases
- this is mitigated but not resolved by a better choice of $\tilde{p}_t(x_t | x_{1:(t-1)})$
- the main reason for particle degeneracy is that particles with low weight at step $t - 1$ tend to have low weight at step t .

At step t , the Monte Carlo estimate of π_t is

$$\hat{\pi}_t(\mathbf{x}) = \sum_{i=1}^N W_t(x_{1:t,i}) \delta_{x_{1:t,i}}(\mathbf{x})$$

This discrete distribution on

$$x_{1:t,1}, \dots, x_{1:t,N}$$

is resampled N times according to the weights

$$W_t(x_{1:t,1}), \dots, W_t(x_{1:t,N}).$$

After resampling, the new resampled values replace the

$$x_{1:t,1}, \dots, x_{1:t,N}$$

collection, and each resampled point carries with it its normalized resampling weight $W_t(x_{1:t,i})$.

Each original value $x_{1:t,i}$ may appear in the resampled collection more than once.

Let $n_{j,1}, \dots, n_{j,N}$ denote the numbers of times that each of the N particles are resampled during multinomial sampling. Let

$$\tilde{\pi}_t(\mathbf{x}) = \sum_{i=1}^N \frac{n_{j,i}}{N} \delta_{x_{1:t,i}}(\mathbf{x})$$

Resampling does not disrupt the particle approximation in expectation (that is, it does not introduce bias), but it does introduce “noise”.

Particle Degeneracy: Resampling

Therefore we perform resampling as infrequently as possible, only when the observed variance of the weights exceeds some threshold.

A statistic to track this is the *effective sample size* (ESS) where

$$\text{ESS} = \left(\sum_{i=1}^N \{W_{t,i}\}^2 \right)^{-1}$$

We have $1 \leq \text{ESS} \leq N$:

$$\text{ESS} = 1 \quad \text{if} \quad W_{t,i} = 1 \quad \text{for some } i$$

$$\text{ESS} = N \quad \text{if} \quad W_{t,i} = \frac{1}{N} \quad \text{for each } i$$

Example: AR(1) plus noise

$$Y_t = X_t + V_t$$

$$X_{t+1} = \phi X_t + W_t$$

where $-1 < \phi < 1$, with $V_t \sim \text{Normal}(0, \sigma_V^2)$ and $W_t \sim \text{Normal}(0, \sigma_W^2)$.

$$p(y_t | x_t) = \text{Normal}(x_t, \sigma_V^2)$$

$$p(x_{t+1} | x_t) = \text{Normal}(\phi x_t, \sigma_W^2)$$

In this case, we have that

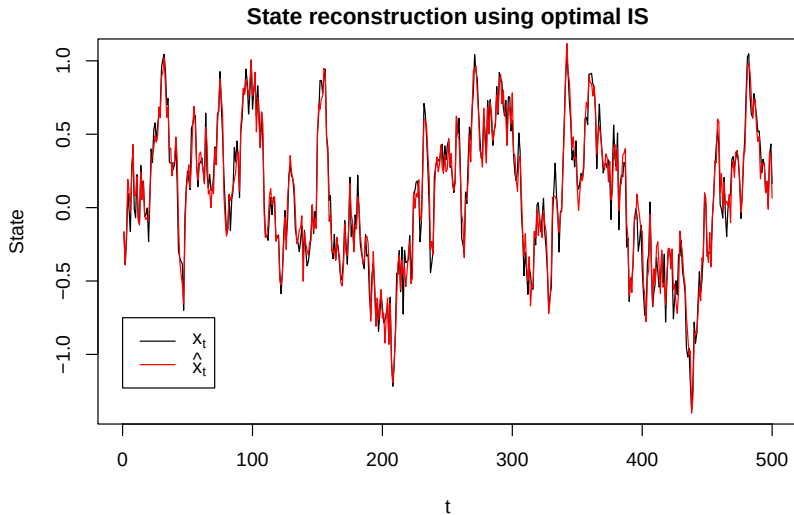
$$\tilde{p}_t(x_t|y_t, x_{(t-1)}) \propto \exp \left\{ -\frac{1}{2\sigma_V^2} (y_t - x_t)^2 \right\} \exp \left\{ -\frac{1}{2\sigma_W^2} (x_t - \phi x_{t-1})^2 \right\}$$

which simplifies to

$$\tilde{p}_t(x_t|y_t, x_{(t-1)}) \equiv \text{Normal} \left(\frac{y_t/\sigma_V^2 + \phi x_{t-1}/\sigma_W^2}{1/\sigma_V^2 + 1/\sigma_W^2}, \left[\frac{1}{\sigma_V^2} + \frac{1}{\sigma_W^2} \right]^{-1} \right)$$

Therefore, we can sample forward directly from $\tilde{p}_t(x_t|y_t, x_{(t-1)})$, and the IS weights are always equal to $1/N$. This calculation follows by the exact same arguments that justify the Kalman Filter recursions.

Example



Now we use the alternative proposal

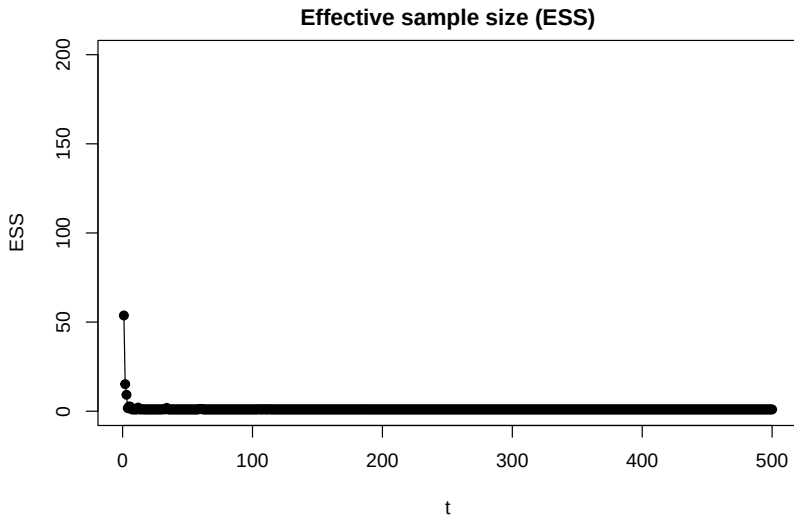
$$\tilde{f}_j(x_j|x_{1:(j-1)}) = p_X(x_j|x_{j-1}) \equiv \text{Normal}(\phi x_{j-1}, \sigma_W^2)$$

which yields the recursive (unnormalized) weight calculation

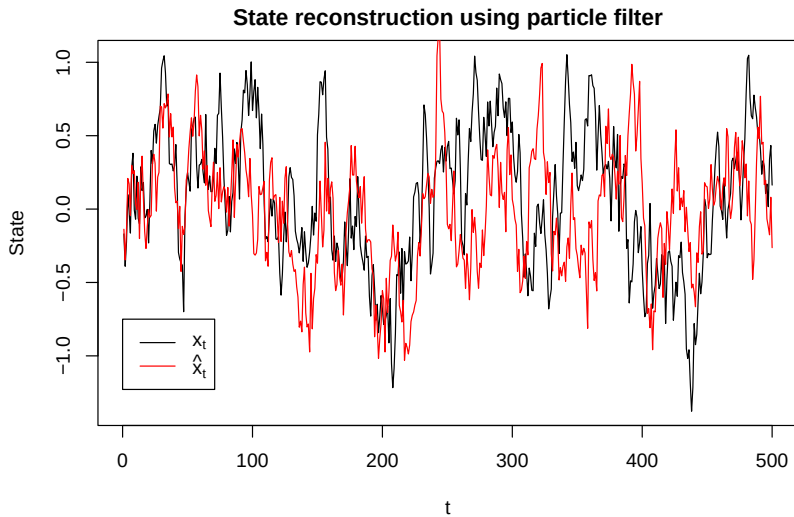
$$w_j(x_{1:j}) = w_{j-1}(x_{1:(j-1)}) \exp \left\{ -\frac{1}{2\sigma_V^2} (y_j - x_j)^2 \right\}.$$

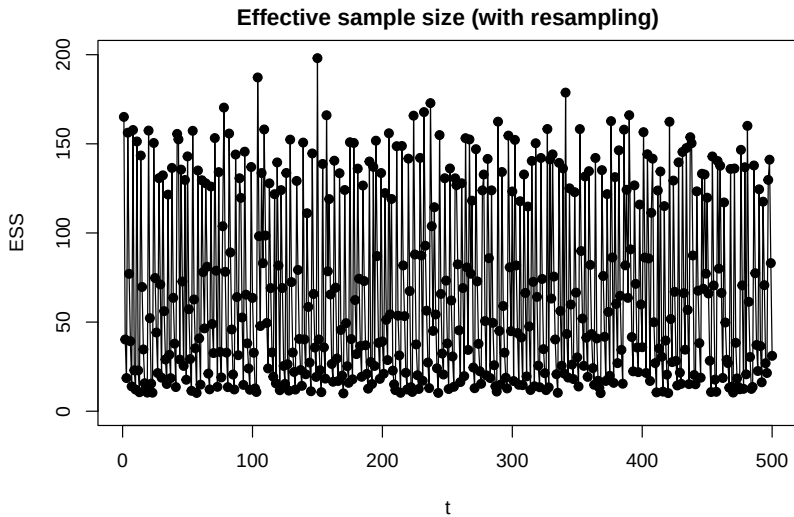
In this case, there is numerical underflow in this weight calculation, so for each j the weights are scaled.

Example

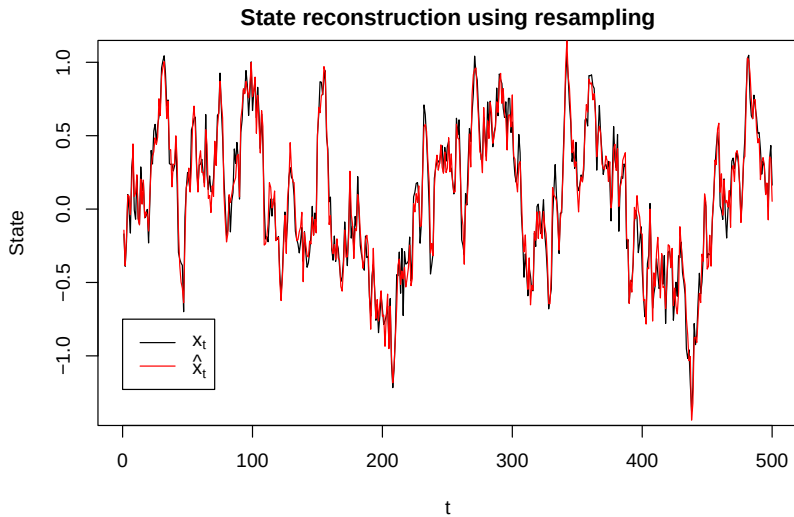


Example





Example



Non-linear example

Consider the non-linear model

$$p_{Y|X}(y_t|x_t) \equiv \text{Normal}(0, \exp\{2x_t\})$$

$$p_X(x_t|x_{t-1}) \equiv \text{Normal}(\mu + \phi(x_{t-1} - \mu), \sigma^2)$$

with $p_{X_1}(x) \equiv \text{Normal}(\mu, \sigma^2/(1 - \phi^2))$.

This model is a non-linear state space model because of the distribution of y_t on x_t .

Here, for $t \geq 2$,

$$p_{Y|X}(y_t|x_t) = \left(\frac{e^{-2x_t}}{2\pi}\right)^{1/2} \exp\{-e^{-2x_t}y_t^2/2\}$$

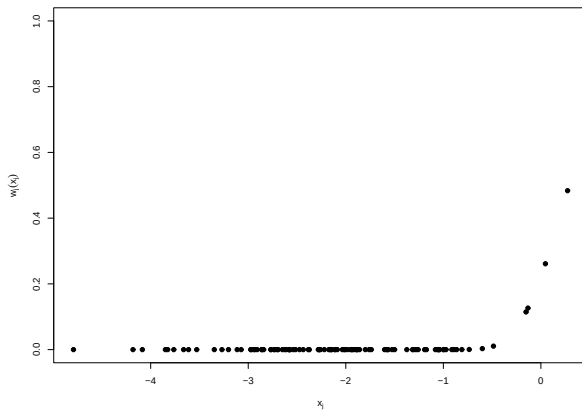
and the recursive weight calculation is based on

$$w_t(x_{1:t}) = \prod_{s=1}^t \frac{\tilde{\pi}_s(x_s|x_{1:(s-1)})}{\tilde{p}_s(x_s|x_{1:(s-1)})} = \prod_{s=1}^t p_{Y|X}(y_s|x_s)$$

that is, the likelihood up to observation t .

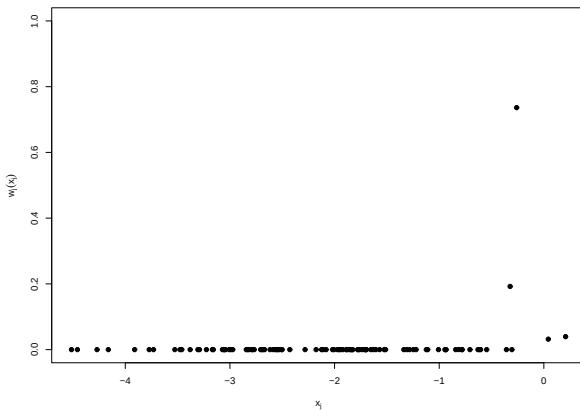
Non-linear example

$N = 100$ particles and weights at $t = 1$



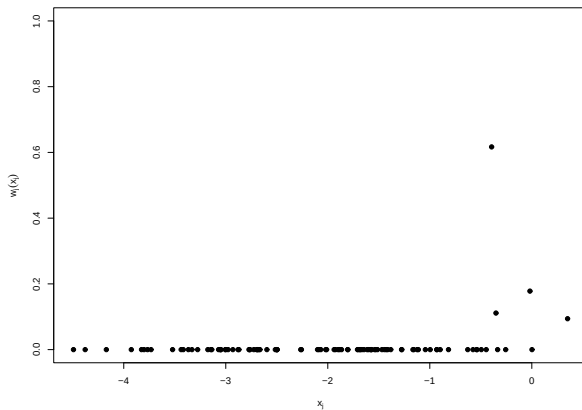
Non-linear example

$N = 100$ particles and weights at $t = 2$



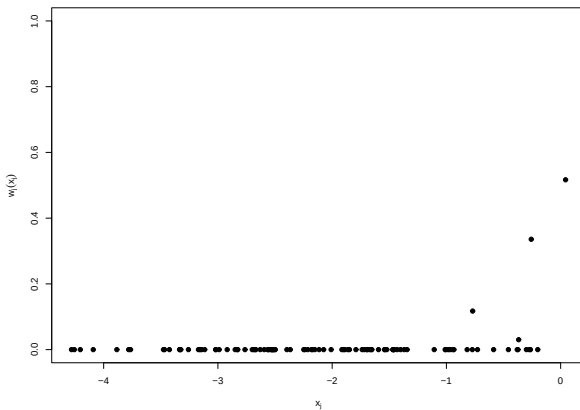
Non-linear example

$N = 100$ particles and weights at $t = 3$



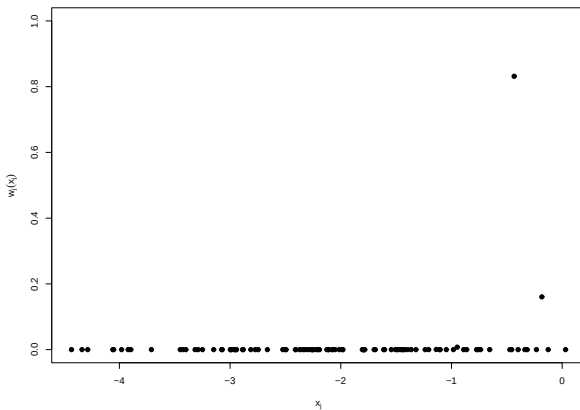
Non-linear example

$N = 100$ particles and weights at $t = 4$



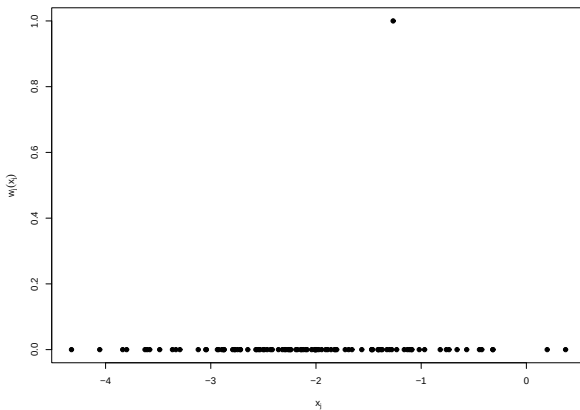
Non-linear example

$N = 100$ particles and weights at $t = 5$



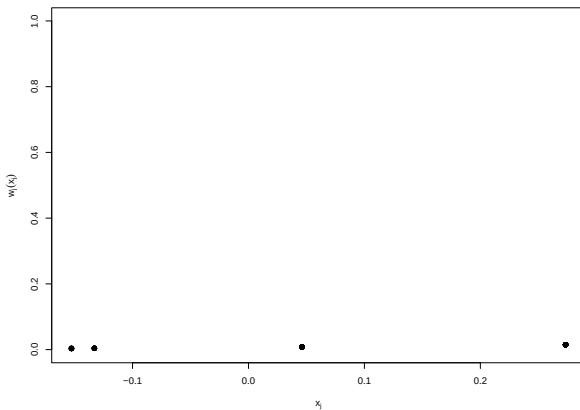
Non-linear example

$N = 100$ particles and weights at $t = 20$



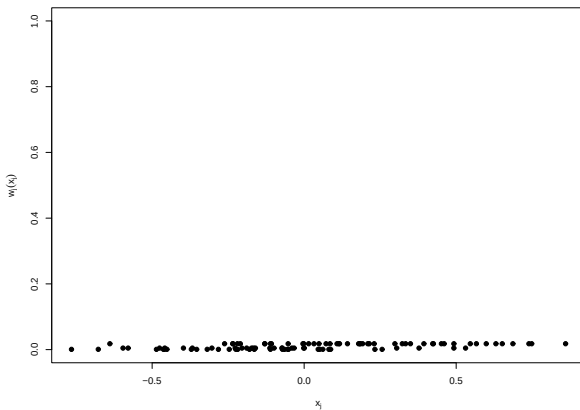
Non-linear example

$N = 100$ resampled particles and weights at $t = 1$



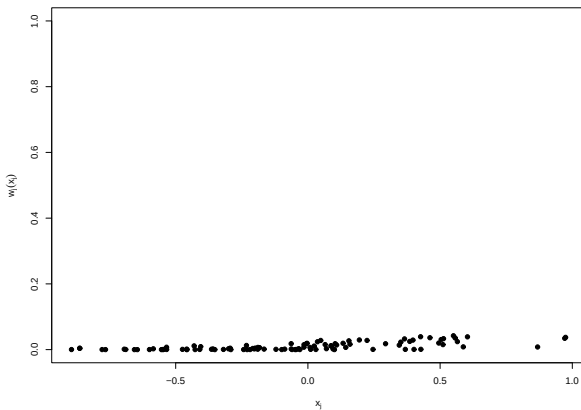
Non-linear example

$N = 100$ resampled particles and weights at $t = 2$



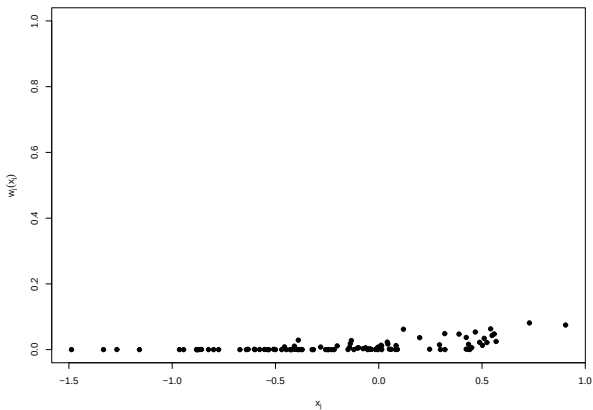
Non-linear example

$N = 100$ resampled particles and weights at $t = 3$



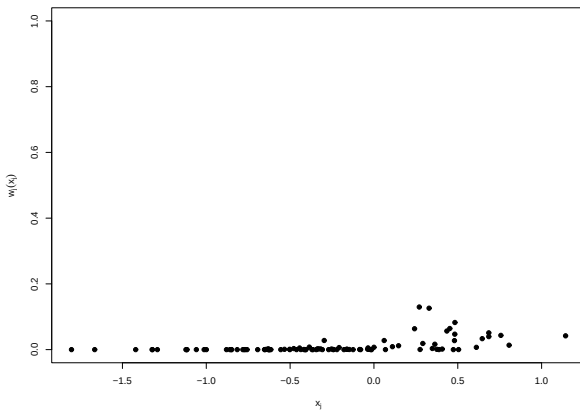
Non-linear example

$N = 100$ resampled particles and weights at $t = 4$



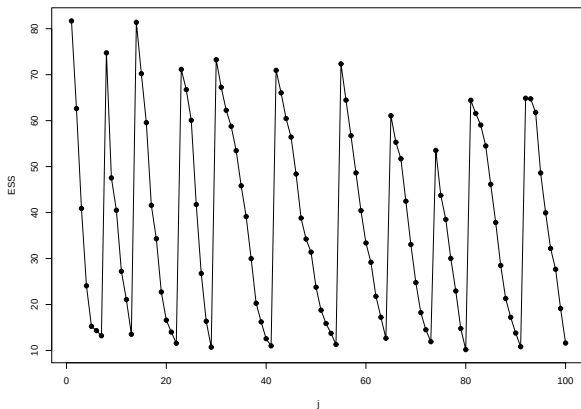
Non-linear example

$N = 100$ resampled particles and weights at $t = 5$

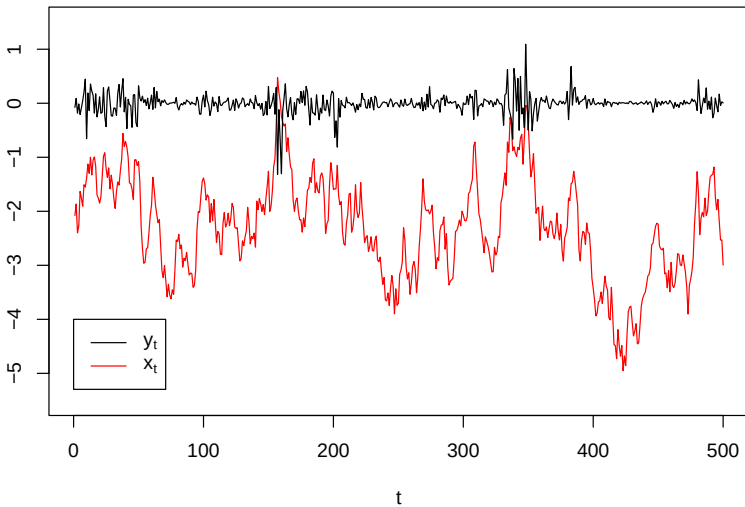


Non-linear example

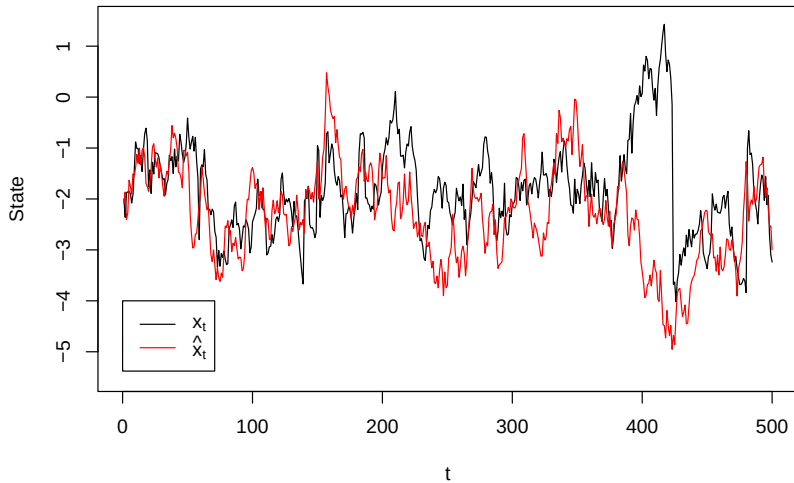
Effective sample size (ESS) with resampling when $ESS < 10$

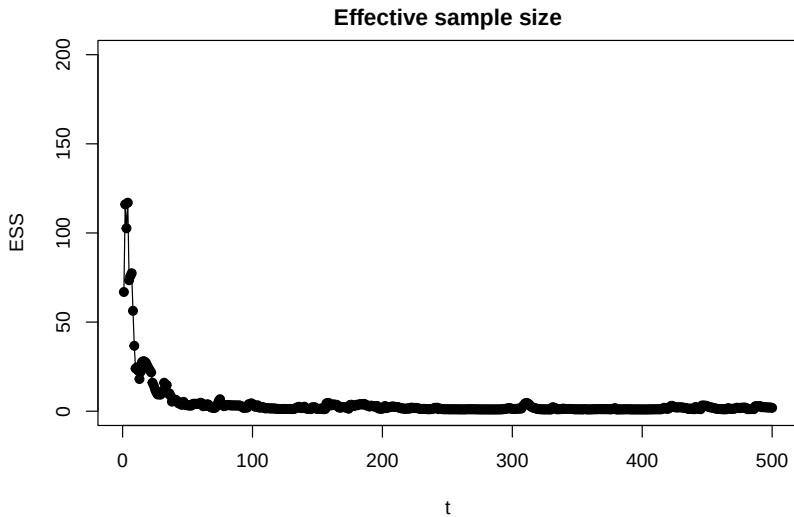


Non-linear example



Non-linear example





Here the mode of

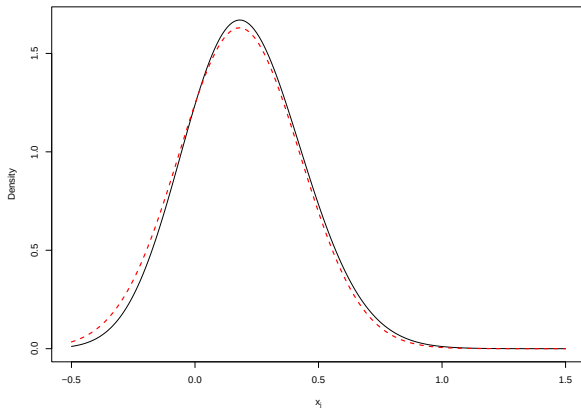
$$p(x|y_t, x_{t-1}) = \exp \left\{ -x - \frac{y_t^2}{2e^{2x}} - \frac{1}{2\sigma^2}(x - \mu_t)^2 \right\}$$

can be computed numerically, and the variance of the normal approximation can be computed by inspecting the curvature at the mode, given by

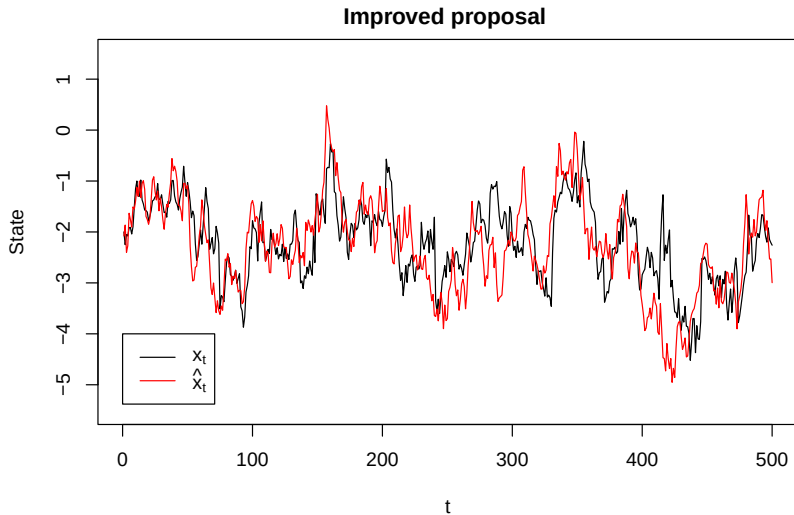
$$\left[-\frac{\partial^2 \log p(x|y_t, x_{t-1})}{\partial x^2} \right]^{-1} = [2y_t^2 e^{-2x} + 1/\sigma^2]^{-1}$$

Non-linear example

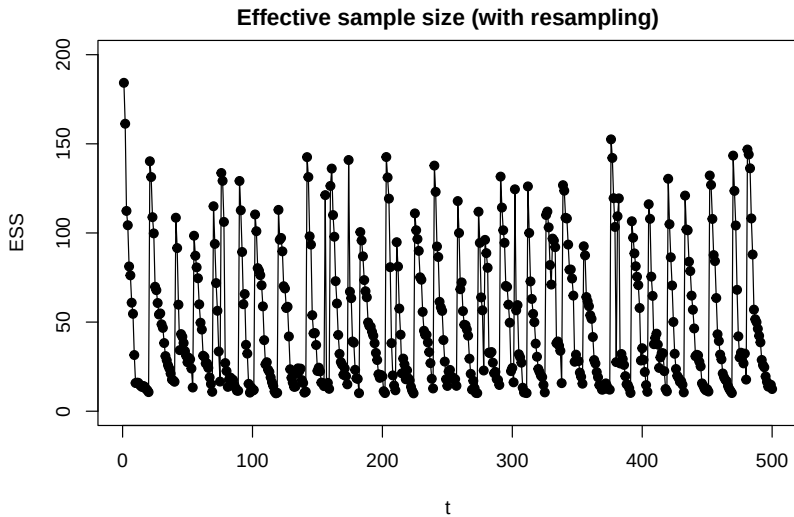
Normal approximation to $p(x_t|y_t, x_{t-1})$ for $x_{t-1} = 0, y_t = 2$:



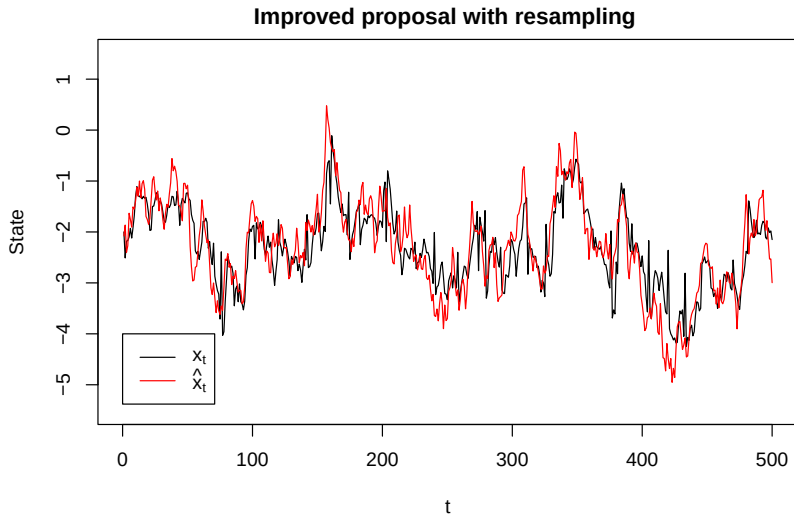
Non-linear example



Non-linear example



Non-linear example

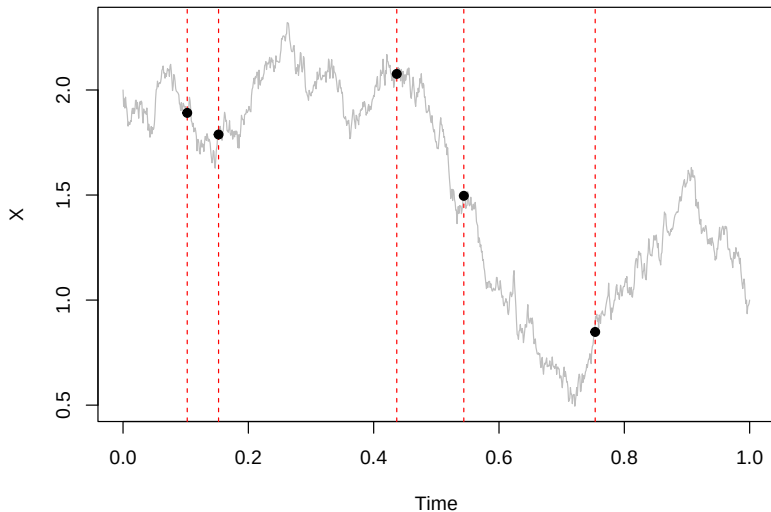


For more general diffusion models and inference tasks, more complex methods are needed.

The two main challenges are

- (a) the need for continuous reconstruction
- (b) unequally spaced observations

More general diffusion models



Note: The strategy of using increasingly fine time-discretization is *not recommended*

- increases the dimension of the augmentation space
- increases the dependence between the discretization points
- MCMC based solutions that use the augmented space suffer critical slowing

For discretely-observed diffusions, it is often not necessary to *prospectively* sample the diffusion path, that is, sample $\{X(t), 0 < t < T\}$ in its entirety.

- only need to sample $X(t)$ at the observation times
- can sample the connecting paths conditionally on the finite number of observation times as required (this step is not trivial).

For some diffusion-based models, this *retrospective* sampling can be performed efficiently.

A number of strategies have been developed:

- most use some form of augmentation, and Monte Carlo approximation of the transition density

$$p(x_t|x_s)$$

- analytical/numerical approximations have been proposed
- neural network emulations also exist

Part 5

Particle Markov chain Monte Carlo

The Bayesian Computation: recap

We make inference about $\psi = (\theta, \phi)$ via the posterior distribution

$$\pi_n(\theta, \phi | \mathbf{y}) = \frac{f(\mathbf{y}; \theta, \phi) \pi_0(\theta, \phi)}{\int f(\mathbf{y}; t, z) \pi_0(t, z) dt dz}$$

where

- θ is the parameter of interest
- ϕ is a nuisance parameter.

that is, our principal focus is the marginal density

$$\pi_n(\theta | \mathbf{y}) = \frac{\int f(\mathbf{y}; \theta, z) \pi_0(\theta, z) dz}{\int \int f(\mathbf{y}; t, z) \pi_0(t, z) dt dz}$$

Example: Log odds ratio

Suppose

- Likelihood: $Y_1|\omega_1 \sim \text{Binomial}(m_1, \omega_1)$, $Y_2|\omega_2 \sim \text{Binomial}(m_2, \omega_2)$ independent.
- Prior: $\omega_1, \omega_2 \sim \text{Beta}(\alpha, \beta)$ independent.

Parameter of interest is

$$\theta = \log \left(\frac{\omega_2/(1 - \omega_2)}{\omega_1/(1 - \omega_1)} \right)$$

Posterior samples of ω_1 and ω_2 can be transformed into a sample of θ .

1. Construct the Markov chain on the *joint* posterior for (θ, ϕ) , and store the sampled values for θ . Acceptance probability is

$$\alpha((\theta, \phi), (\theta', \phi')) = \min \left\{ 1, \frac{\pi_n(\theta', \phi' | \mathbf{y})}{\pi_n(\theta, \phi | \mathbf{y})} \frac{q((\theta, \phi), (\theta', \phi'))}{q((\theta', \phi'), (\theta, \phi))} \right\}.$$

2. Construct the Markov chain on the *marginal* for θ . Acceptance probability is

$$\alpha(\theta, \theta') = \min \left\{ 1, \frac{\pi_n(\theta' | \mathbf{y})}{\pi_n(\theta | \mathbf{y})} \frac{q(\theta, \theta')}{q(\theta', \theta)} \right\}.$$

The difficulty in implementing 2. is that perhaps the marginal posterior is not available analytically, so that the Hastings ratio and acceptance probability cannot be computed.

- $q(\psi, \psi')$ is some appropriately chosen proposal density
- we only need $\pi_n(\theta, \phi|\mathbf{y})$ or $\pi_n(\theta|\mathbf{y})$ up to proportionality; the normalizing constant cancels in the Hastings ratio.

$$\pi_n(\theta|\mathbf{y}) = \frac{\int f(\mathbf{y}; \theta, z) \pi_0(\theta, z) dz}{\int \int f(\mathbf{y}; t, z) \pi_0(t, z) dt dz} \propto \int f(\mathbf{y}; \theta, z) \pi_0(\theta, z) dz$$

If (θ, ϕ) are *a priori* independent

$$\pi_n(\theta|\mathbf{y}) \propto \left\{ \int f(\mathbf{y}; \theta, z) \pi_0(z) dz \right\} \pi_0(\theta) = \mathcal{L}_n(\theta; \mathbf{y}) \pi_0(\theta)$$

say.

Can approximate $\mathcal{L}_n(\theta; \mathbf{y})$ by *Monte Carlo*

$$\hat{\mathcal{L}}_n(\theta; \mathbf{y}) = \frac{1}{N} \sum_{i=1}^N f(\mathbf{y}; \theta, \phi^{(i)})$$

where $\{\phi^{(i)}\}$ are i.i.d from $\pi_0(\phi)$, or *Importance Sampling*

$$\hat{\mathcal{L}}_n(\theta; \mathbf{y}) = \frac{1}{N} \sum_{i=1}^N \frac{f(\mathbf{y}; \theta, \phi^{(i)}) \pi_0(\phi^{(i)})}{f_0(\phi^{(i)})}$$

where $\{\phi^{(i)}\}$ are i.i.d. from $f_0(\phi)$, where f_0 is an IS density.

For all θ , $\hat{\mathcal{L}}_n(\theta; \mathbf{y})$ is an *unbiased* estimator of $\mathcal{L}_n(\theta; \mathbf{y})$.

Example: Log odds ratio

$$\theta = \log \left(\frac{\omega_2/(1 - \omega_2)}{\omega_1/(1 - \omega_1)} \right) \quad \phi = \omega_1$$

so that

$$\pi_n(\theta, \phi | y_1, y_2) \propto Be_1(\phi) Be_2((\phi e^\theta)/(1 - \phi + \phi e^\theta)) \frac{\phi(1 - \phi)e^\theta}{(1 - \phi + \phi e^\theta)^2}$$

on $\mathbb{R} \times [0, 1]$, where

- Be_1 is the $Beta(y_1 + \alpha, m_1 - y_1 + \beta)$ pdf
- Be_2 is the $Beta(y_2 + \alpha, m_2 - y_2 + \beta)$ pdf

Why not just use $\widehat{\mathcal{L}}_n(\theta; \mathbf{y})$ in place of $\mathcal{L}_n(\theta; \mathbf{y})$ in the marginal Metropolis-Hastings ? Acceptance probability is

$$\alpha(\theta, \theta') = \min \left\{ 1, \frac{\widehat{\mathcal{L}}_n(\theta'; \mathbf{y}) \pi_0(\theta')}{\widehat{\mathcal{L}}_n(\theta; \mathbf{y}) \pi_0(\theta)} \frac{q(\theta', \theta)}{q(\theta, \theta')} \right\}$$

With N large, this should give a reasonable approximation to the correct answer (ie the chain will converge to more or less the correct target).

- $\widehat{\mathcal{L}}_n(\theta; \mathbf{y})$ is *retained* from the current computation (at θ),
- $\widehat{\mathcal{L}}_n(\theta'; \mathbf{y})$ is computed afresh (at θ').

It transpires that the stationary distribution of the chain implied by the naive construction (using $\hat{\mathcal{L}}_n$) *is the target marginal distribution*.

- for any N , the chain produces samples from the marginal for the parameter of interest.
- this does not solve all implementation issues, such as
 - ▶ choosing N
 - ▶ choosing q
 - ▶ choosing amongst the many variants of this approach.

Example: Log odds ratio

$$\pi_n(\theta, \phi | y_1, y_2) \propto Be_1(\phi) Be_2((\phi e^\theta)/(1 - \phi + \phi e^\theta)) \\ \times \frac{\phi(1 - \phi)e^\theta}{(1 - \phi + \phi e^\theta)^2}$$

on $\mathbb{R} \times [0, 1]$.

Example: Log odds ratio

- Initialization at $\theta^{(0)}$; compute $\hat{\mathcal{L}}(\theta^{(0)}; \mathbf{y})$ by MC/IS.
- for iteration $i = 1, 2, \dots$
 - ▶ propose $\theta' \sim q(\theta^{(i)}, \cdot)$.
 - ▶ compute $\hat{\mathcal{L}}(\theta'; \mathbf{y})$.
 - ▶ perform MH step: accept move to $\theta^{(i+1)} = \theta'$ with probability $\alpha(\theta^{(i)}, \theta')$. Otherwise set $\theta^{(i+1)} = \theta^{(i)}$.

If move is accepted, retain $\hat{\mathcal{L}}(\theta'; \mathbf{y})$, otherwise retain $\hat{\mathcal{L}}(\theta^{(t)}; \mathbf{y})$.

Example: Log odds ratio

With q taken to be the Gaussian Metropolis kernel with parameter σ , the following effective sample sizes are obtained for the marginal algorithm after 5000 iterations (measure is percent).

N	σ										
	0.1	0.2	0.3	0.4	0.5	1.0	1.5	2.0	3.0	4.0	5.0
50	0.46	1.41	2.24	2.85	5.58	13.57	17.35	17.56	13.39	12.23	7.92
100	0.58	1.80	2.88	5.23	7.77	15.68	20.45	16.32	14.67	10.94	11.26
200	0.49	1.58	3.18	4.86	6.96	17.15	21.78	20.10	16.47	15.77	11.18
500	0.53	1.96	3.22	4.15	6.37	16.96	20.66	19.37	13.48	18.04	10.53
1000	0.31	1.84	3.49	5.31	8.55	14.87	16.96	22.13	18.01	12.18	12.47
2000	0.64	1.46	3.16	5.23	8.19	16.86	20.40	20.93	17.37	15.21	12.53
5000	0.44	1.60	2.77	4.60	7.28	17.10	20.85	21.78	16.53	11.83	11.53
10000	0.58	1.37	3.32	5.36	8.56	18.48	23.17	24.98	16.08	14.42	11.85

This method has been deployed successfully for the state space models and other models based on stochastic processes as the filtering-based SMC estimator

$$\hat{\mathcal{L}}_n(\theta; \mathbf{y}) = \frac{1}{N} \sum_{i=1}^N \left\{ \prod_{t=1}^n \hat{p}(y_t | y_{1:(t-1)}; \theta) \right\}$$

is an unbiased estimator of $\mathcal{L}_n(\theta; \mathbf{y})$. Here, the ϕ parameters are the particle collections representing the latent states.

The R package `pmhtutorial` contains one accessible implementation for the SV model.

Note: this method

- is based on a remarkable theoretical result
- is *extremely* computationally intensive if n is large
- might only be used as a last resort

Part 6

Hidden Markov Models

6.1. Hidden Markov Models

If the unobserved state X_t is *discrete*, then state space models are often called *hidden Markov models* (HMMs).

In this case, the state equation is replaced by a *Markov chain* model: if $\{X_t\}$ can take values on $\{1, 2, \dots, S\}$ say, then for $t > 1$

$$p(x_t | x_{1:(t-1)}) \equiv p(x_t | x_{t-1})$$

where

$$\Pr[X_t = l | X_{t-1} = k] = p_{kl}$$

The $S \times S$ transition matrix $\mathbf{P}$, along with the (marginal) distribution of X_1 ($\Pr[X_1 = k], k = 1, \dots, S$) characterize the nature of the state evolution. Note that $\mathbf{P}$ is a *stochastic matrix* whose rows sum to 1.

All of the previous types of calculation

- filtering $p(x_t|y_{1:t})$
- prediction $p(x_t|y_{1:(t-1)})$
- smoothing $p(x_t|y_{1:n})$
- parameter estimation

are required. The main difference is that the filtering, prediction and smoothing distributions are *discrete*, and therefore more straightforward to compute.

The observation variable y_t can be either discrete or continuous.

6.1. Hidden Markov Models

Example: $S = 2$

Suppose for $t = 1, 2, \dots, n$

$$Y_t = \mu(X_t) + \epsilon_t$$

and $\{X_t\}$ follows a Markov chain with transition matrix

$$\mathbf{P} = \begin{bmatrix} 0.3 & 0.7 \\ 0.4 & 0.6 \end{bmatrix}$$

and the distribution of X_1 is the stationary (or equilibrium) distribution, π , of the chain defined as the solution of

$$\pi = \pi \mathbf{P}$$

6.1. Hidden Markov Models

Example: $S = 2$

We compute $\boldsymbol{\pi}$ (a 1×2 vector) by

$$[1, 1]^T \boldsymbol{\pi} = \lim_{t \rightarrow \infty} \mathbf{P}^t = \begin{bmatrix} 4/11 & 7/11 \\ 4/11 & 7/11 \end{bmatrix}$$

so $\boldsymbol{\pi} = (4/11, 7/11)$.

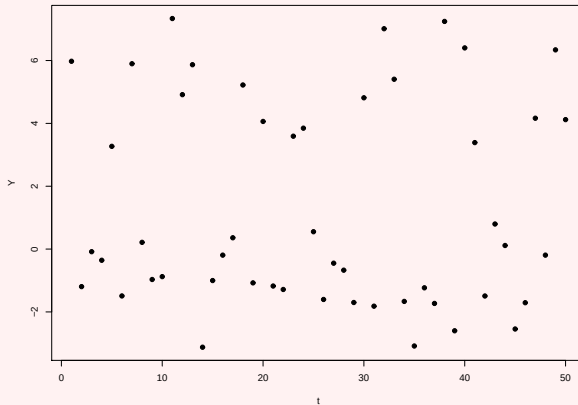
Suppose

$$\mu(x) = \begin{cases} \mu_1 & x = 1 \\ \mu_2 & x = 2 \end{cases}$$

and $\{\epsilon_t\} \sim \text{Normal}(0, 1)$, IID.

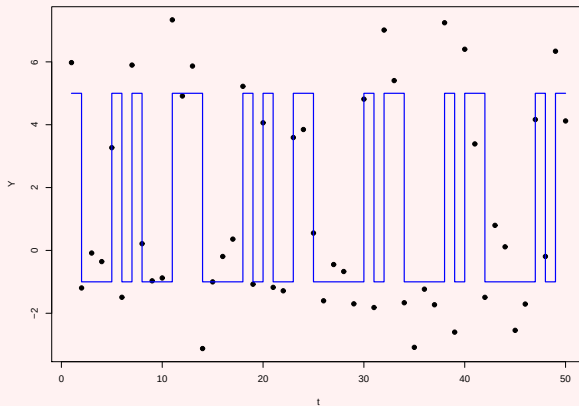
6.1. Hidden Markov Models

Example: $S = 2$



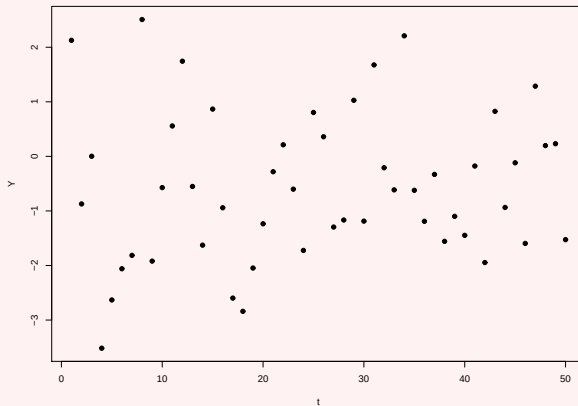
6.1. Hidden Markov Models

Example: $S = 2, \mu_1 = 5, \mu_2 = -1$



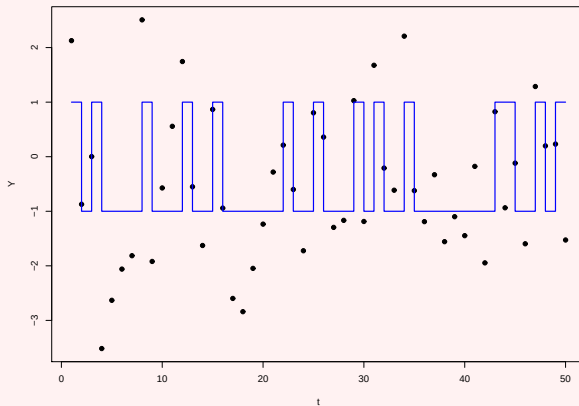
6.1. Hidden Markov Models

Example: $S = 2$



6.1. Hidden Markov Models

Example: $S = 2, \mu_1 = 1, \mu_2 = -1$



6.1. Hidden Markov Models

Filtering: By Bayes theorem and the HMM structure

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t)p(x_t|y_{1:(t-1)})}{\sum_{x=1}^S p(y_t|x)p(x|y_{1:(t-1)})}$$

This is a discrete distribution on $\{1, \dots, S\}$ that we can compute exactly.

Prediction: By the theorem of total probability

$$p(x_{t+1}|y_{1:t}) = \sum_{x=1}^S p(x_{t+1}|x)p(x|y_{1:t})$$

- $p(x_{t+1}|x)$ is determined the x th row of $\mathbf{P}$, for each $x = 1, 2, \dots, S$
- $p(x|y_{1:t})$ is the previous filtering distribution.

6.1. Hidden Markov Models

Smoothing: Computing *backwards* $t = n, n - 1, \dots, 1$,

$$p(x_t | y_{1:n}) = \sum_{x_{t+1}=1}^S p(x_t | x_{t+1}, y_{1:n}) p(x_{t+1} | y_{1:n})$$

$p(x_{t+1} | y_{1:n})$ is the previous smoothing distribution. For the first term

$$\begin{aligned} p(x_t | x_{t+1}, y_{1:n}) &\equiv p(x_t | x_{t+1}, y_{1:t}) = \frac{p(x_t | y_{1:t}) p(x_{t+1} | x_t, y_{1:t})}{p(x_{t+1} | y_{1:t})} \\ &= \frac{p(x_t | y_{1:t}) p(x_{t+1} | x_t)}{p(x_{t+1} | y_{1:t})} \end{aligned}$$

which can be computed from the filtering and prediction distributions.

In the discrete state case, all of the relevant distributions for inference

- $p(x_t|y_{1:n}; \theta)$
- $p(y_t|y_{1:(t-1)}; \theta)$

are available numerically, so sampling the posterior for the latent states

$$\pi_n(x_{1:n}|y_{1:n}, \theta)$$

exactly is straightforward. Sampling

$$\pi_n(\theta|x_{1:n}, y_{1:n})$$

typically requires Metropolis-Hastings.

In this case also, we have access to the *marginal* posterior

$$\pi_n(\theta|y_{1:n}) \propto \mathcal{L}_n(\theta)\pi_0(\theta)$$

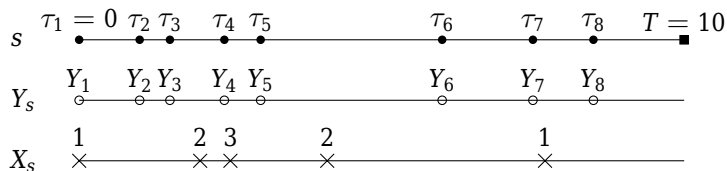
as

$$\mathcal{L}_n(\theta) = p(y_1; \theta) \prod_{t=2}^n p(y_t|y_{1:(t-1)}; \theta)$$

so we can run the Metropolis-Hastings algorithm on the marginal model.

Continuous Time Markov Models

In *continuous time* the calculation is more complex.



The process $\{X(t)\}$ follows a continuous time Markov chain (CTMC).

$\{X(t)\}$ comprises an uncountable collection of random variables that take values on $\mathcal{X}$, subject to the Markov assumption.

For all $s, t > 0$, and for all j and k

$$P[X(t+s) = k | X(s) = j, \mathcal{H}_s] = P[X(t+s) = k | X(s) = j]$$

where $\mathcal{H}_s$ is (informally) the history of the chain up to time s . Typically the chain is presumed to be time homogeneous, so that

$$P[X(t+s) = k | X(s) = j] \equiv P[X(t) = k | X(0) = j] = P_{jk}(t).$$

say, for all s, t, j, k .

We consider the sequence of transition times

$$0 = t_0 < t_1 < \cdots < t_n < \cdots$$

which is a countable collection of non-negative, increasing random variables, along with the countable collection of random variables

$$X(0), X(t_1), X(t_2), \dots, X(t_n), \dots$$

where we might write $X_{t_n} \equiv X(t_n) \equiv X_n$.

Continuous Time Markov Models

The rate (per unit time) of transitions *out* of each state is

$$q_{jk} = \lim_{t \rightarrow 0} \frac{P_{jk}(t)}{t} \quad j \neq k.$$

The *net* (or *aggregate*) rate of transitions out of state j can be written

$$\lambda_j = \sum_{k \neq j} q_{jk}$$

and the *instantaneous* probability of transition by defining

$$P_{jk} = \frac{q_{jk}}{\sum_{i \neq j} q_{ji}}$$

which does not depend on t , and satisfies

$$0 \leq P_{jk} \leq 1 \quad \sum_{k \neq j} P_{jk} = 1.$$

We may observe $\{X(t)\}$ at

- (i) the *exact* transition times between states
- (ii) *arbitrary* time points

For (i), the likelihood is straightforward. For observation window $[0, T]$

$$\prod_{j=1}^K \prod_{k \neq j} q_{j,k}^{N_{j,k}(T)} \exp \{-q_{j,k} R_j(T)\}$$

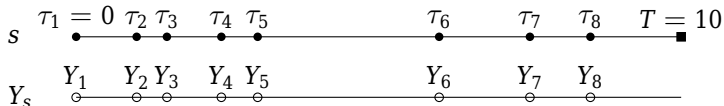
where $N_{j,k}(T)$ is the *number of transitions* from state j to state k , and

$$R_j(T) = \int_0^T \mathbb{1}(X(s) = j) ds$$

is the *total time* that the process has spent in state j in $[0, T]$.

For (ii), the likelihood is computed using $P(t)$, but is less straightforward

- R package `msm` performs maximum likelihood estimation in both models
- `ctmcd` implements a Bayesian MCMC-based analysis.
- we also have the complication of *hidden* versions of this model, where $\{X(s)\}$ is not observed.



For inference, we can deploy the augmentation approach; to sample the conditional posterior $\pi_n(\mathbf{X}|\theta)$

- the states $X_1, X_2, \dots, X_n$ at the *observation* times
- the continuous path segments

$$X(\tau_1) \rightarrow X(\tau_2), X(\tau_2) \rightarrow X(\tau_3), \dots X(\tau_{n-1}) \rightarrow X(\tau_n), X(\tau_n) \rightarrow T$$

- To sample the path segment

$$X(\tau_j) \rightarrow X(\tau_{j+1})$$

conditional on the sampled endpoint states X_j, X_{j+1} requires a rejection sampling step.

This allows a complete reconstruction of the state path for all individuals over the observation window.

Example: Health trajectories

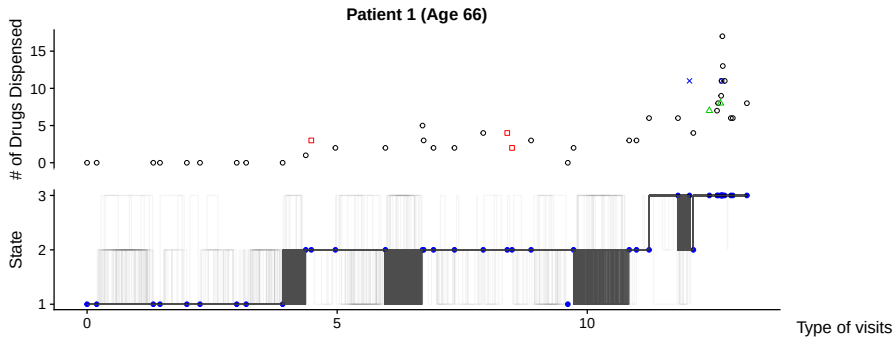
In a large cohort, (81000 COPD patients) we observe

- health system utilization: number of prescriptions prescribed by
 - ▶ primary care visits
 - ▶ hospitalizations
 - ▶ specialists
 - ▶ emergency care visits
- other predictors (eg age at enrollment)

and attempt to infer the underlying health status of the individual.

For COPD, a 4-state model is supported by clinical information.

Inferred state path for one COPD patient:



Luo et al. (2020)

We can extend this basic model to

- consider an *unknown* number of states
- consider cluster structure in the cohort, e.g. M clusters with generator matrices

$$Q_1, Q_2, \dots, Q_M.$$

- informative observation times.

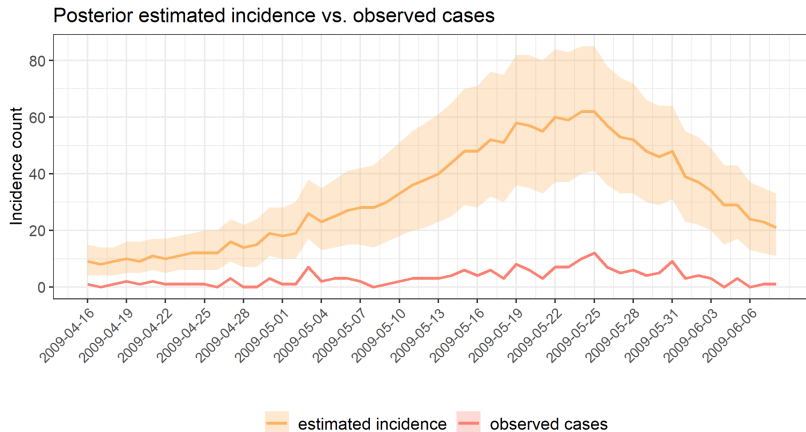
Luo et al. 2020, 2021, 2023.

Part 7

Stochastic Epidemic Models

Stochastic models of infectious disease epidemics

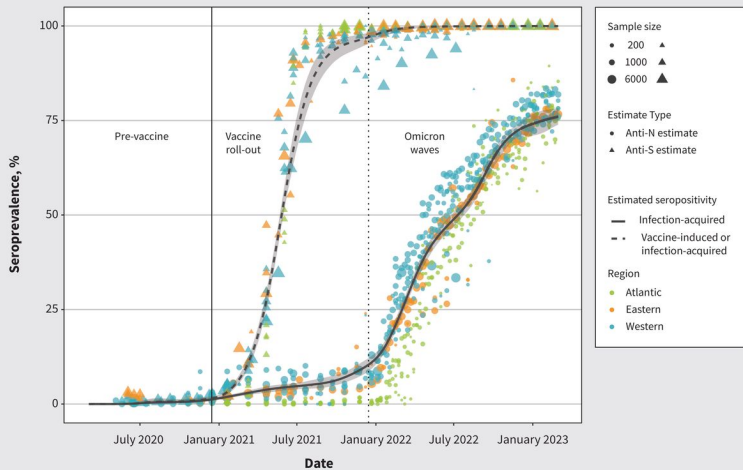
Goal: To estimate how many people have been infected over time and the epidemic characteristics of the infectious disease.



Public health surveillance

- monitoring of developing disease incidence.
- involves synthesizing different data sources to inform public health management and decision making.
- For COVID 19:
 - ▶ case counts
 - ▶ death counts
 - ▶ hospitalizations
 - ▶ test positivity
 - ▶ *seroprevalence*: detection of antibodies in samples from study participants and blood donors

COVID Immunity Task Force (Murphy et al. 2023)



SIR Model: Susceptible-Infected-Recovered

$$\frac{dS(t)}{dt} = -\beta I(t)S(t)$$

$$\frac{dI(t)}{dt} = \beta I(t)S(t) - \gamma I(t)$$

$$\frac{dR(t)}{dt} = \gamma I(t)$$

Infection rate



Recovery rate



Challenges:

- risk stratification by age, region etc.
- parameters β , γ can be time-varying
 - ▶ β related to *reproduction number*
- need to integrate other data sources
- *we do not observe all cases*: unobserved infections
- many time-varying policies
- time-varying immunity

We developed a latent process statistical model to integrate data sources and to reflect time-varying aspects.

Wu et. al. (2024).

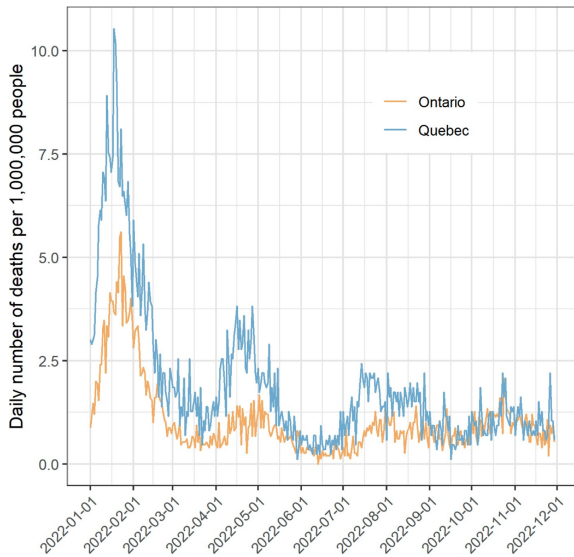
We retain the SIR model, but allow the infection rate parameter to follow a stochastic model, approximating a mean-reverting process.

$$X_{t+1} = \mu + \phi(X_t - \mu) + W_t$$

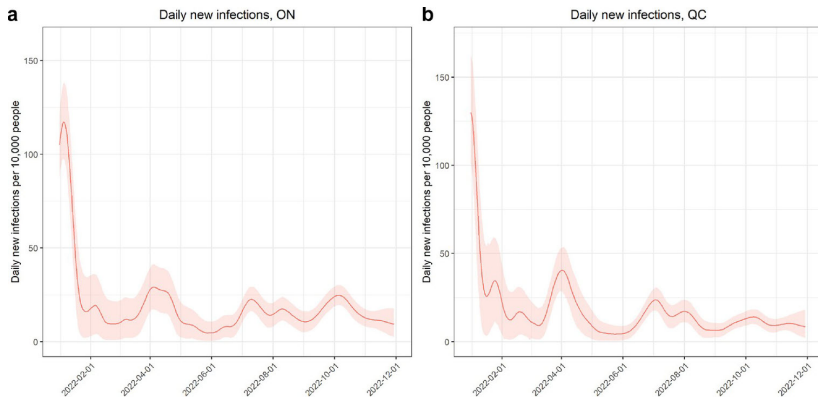
where $X_t = \log \beta_t$.

Here a conventional MCMC strategy implemented in `rstan` suffices. The Bayesian model allows integration of all data sources.

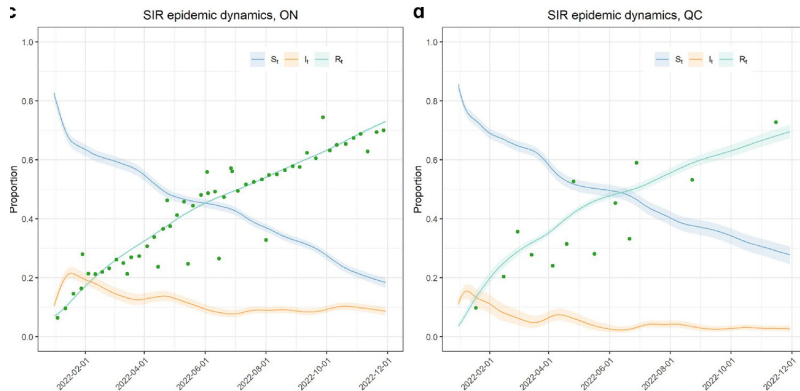
Quebec and Ontario



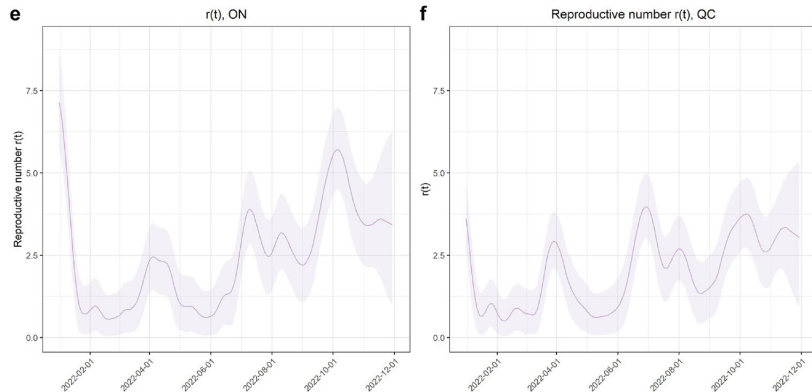
Inferred daily infections.



Inferred SIR incorporating serosurvey data (green dots).



Reproduction number, $r(t)$.



Suppose we retain aspects of, but extend, the SIR model:

- Y_k = reported cases at t_k
- X_k = true incidence counts at t_k
 - ▶ new infections occurring in $(t_{k-1}, t_k]$
- φ = case detection rate
- For $k = 1, \dots, K$,

$$Y_k \mid X_k, \varphi \sim \text{Binomial}(X_k, \varphi).$$

Deterministic models cannot account for the inherent variability in infection and recovery processes.

- adequate for large populations and large outbreaks
- unsuitable for smaller populations

We need *stochastic epidemic models (SEMs)* for outbreaks in small to moderate-sized populations or when the prevalence is low.

The corresponding stochastic process model is a *Markov jump process* (MJP).

- Let $\mathbf{W}(t) = (S(t), I(t))$ denote the count of individuals in the susceptible and infectious state at time t , $t \geq 0$. Let β denote the per-contact infection rate and γ the recovery rate.
- Times between transitions are *Exponentially* distributed with rates

$$\lambda_{(S(t), I(t)), (S(t)+a, I(t)+b)}(\mathbf{W}, t) = \begin{cases} \beta I(t), & (a, b) = (-1, +1) \\ \gamma, & (a, b) = (0, -1) \\ 0, & \text{otherwise.} \end{cases}$$

- The infection rate $\beta I(t)$ at t depends on $I(t)$.
- Fitting the SEM to epidemic trajectories observed at discrete times requires integrating all possible latent continuous-time transition paths - the observed data likelihood is *non-tractable*.

- Discrete-time approximation
- Particle MCMC methods (PMMH)
- *Individual-level data augmentation (DA)*
 - ▶ Augment *individual* epidemic trajectories (i.e., infection and recovery times) to the latent space to have a tractable complete data likelihood.

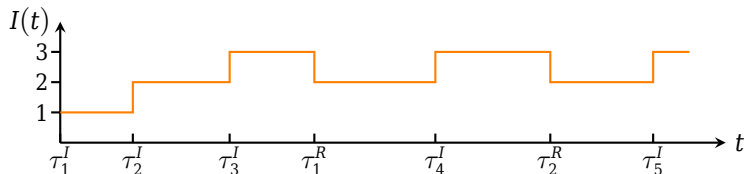
Wu et al. (2025)

- Instead of the aggregate $\mathbf{W}(t)$, we examine the joint process $\mathbf{Z}(t) = (Z_1(t), \dots, Z_P(t))$ of individual sub-processes.
- The subject-level sub-process $Z_j(t)$ is fully characterized by an individual's infection and recovery times τ_j^I and τ_j^R .
- Given current state $\mathbf{Z}(t)$, transition rates for each sub-process are

$$\lambda_{Z_j, Z'_j}(\mathbf{Z}, t) = \begin{cases} \beta I(t), & Z_j = S, Z'_j = I \\ \gamma, & Z_j = I, Z'_j = R \\ 0, & \text{otherwise.} \end{cases}$$

What can we learn from $\mathbf{Z}(t)$?

- If we sort all event times τ along the time line,



then within each inter-event interval,

- ▶ the infection rate $\beta I(t)$ is constant;
- ▶ the likelihood for the individual experiencing the event is the pdf of the Exponential distribution;
- ▶ the likelihood for individuals not experiencing the event is the survival probability - memoryless property.

Let $\boldsymbol{\tau} = \{\tau_1, \dots, \tau_L\}$ denote the (ordered) transition event times of all individuals during the observation period $[0, T]$.

The complete data likelihood is

$$\begin{aligned}\mathcal{L}_n(\mathbf{Z}, \theta) &= f(\mathbf{Y} \mid \mathbf{Z}, \varphi) p(\mathbf{Z} \mid \beta, \gamma) \\&= \prod_{k=1}^K \binom{X_k}{Y_k} \varphi^{Y_k} (1 - \varphi)^{X_k - Y_k} \\&\quad \times \prod_l (\beta I(\tau_l) \mathbb{1}_{\{\tau_l \in \boldsymbol{\tau}^I\}} + \gamma \mathbb{1}_{\{\tau_l \in \boldsymbol{\tau}^R\}}) \\&\quad \exp\{-(\tau_l - \tau_{l-1})(\beta I(\tau_{l-1})S(\tau_{l-1}) + \gamma I(\tau_{l-1}))\} \\&\quad \times \exp\{-(T - \tau_L)(\beta I(\tau_L)S(\tau_L) + \gamma I(\tau_L))\}.\end{aligned}$$

The posterior is

$$\pi_n(\theta, \mathbf{Z} \mid \mathbf{Y}) \propto \mathcal{L}_n(\mathbf{Z}, \theta) \pi(\theta),$$

and we can alternate sampling from $\pi(\theta \mid \mathbf{Y}, \mathbf{Z})$ and $\pi(\mathbf{Z} \mid \mathbf{Y}, \theta)$ using a Gibbs sampler strategy.

The main challenge is to efficiently sample from $\pi(\mathbf{Z} \mid \mathbf{Y}, \theta)$, that is, to efficiently propose and update the latent individual event times.

- Per-subject sampling: very slow and poor MCMC mixing.
- *Block sampling*:
 - ▶ Jointly propose and accept/reject latent event times in blocks.

Recall $\mathbf{X} = (X_1, \dots, X_K)$ is the true incidence counts at each observation time.

- Conditioning on $\mathbf{X}$, we can naturally split all individuals into $K + 1$ infection sets $\mathcal{I}_k, k = 0, \dots, K$, and the set $\mathcal{S}_K$, in which individuals remain susceptible.
- The k -th block consists of the infection and recovery times of individuals in infection set $\mathcal{I}_k$.
- But $\mathbf{X}$ is random, so the size of each block can change.
- We propose to further augment $\mathbf{X}$ to the latent variable space, so that the augmented complete likelihood is

$$\mathcal{L}_n(\mathbf{Z}, \mathbf{X} \mid \theta) = f(\mathbf{Y} \mid \mathbf{X}, \varphi) \delta_{\mathbf{X}}(\mathbf{Z}) p(\mathbf{Z} \mid \beta, \gamma).$$

and we alternate sampling from $\pi(\theta \mid \mathbf{Z}, \mathbf{Y})$ and $\pi(\mathbf{Z}, \mathbf{X} \mid \mathbf{Y}, \theta)$.

In each Metropolis-Hastings step, we can sequentially simulate

- X'_k given $(\mathbf{Z}_{(-k)}, \mathbf{X}_{(-k)}, \mathbf{y})$
- $\mathbf{Z}'_k$ given $(X'_k, \mathbf{X}_{(-k)}, \mathbf{Z}_{(-k)})$, to construct a proposal for $(\mathbf{Z}_k, X_k)$.
- Exact simulation of event times according to the inhomogeneous CTMC is possible but not efficient.
- *Discrete-time approximation* to the MJP to have transition rates constant over each inter-observation interval.
 - ▶ know transition density of X_k
 - ▶ event times $\mathbf{Z}_k = \{(\tau_j^I, \tau_j^R) : j = 1, \dots, x_k\}$ have (truncated) Exponential distributions with identical rates

Extending the SIR to SEIR (including an Exposed state)

$$\frac{dS(t)}{dt} = -\beta I(t)S(t)$$

$$\frac{dE(t)}{dt} = \beta I(t)S(t) - \alpha E(t)$$

$$\frac{dI(t)}{dt} = \alpha E(t) - \gamma I(t)$$

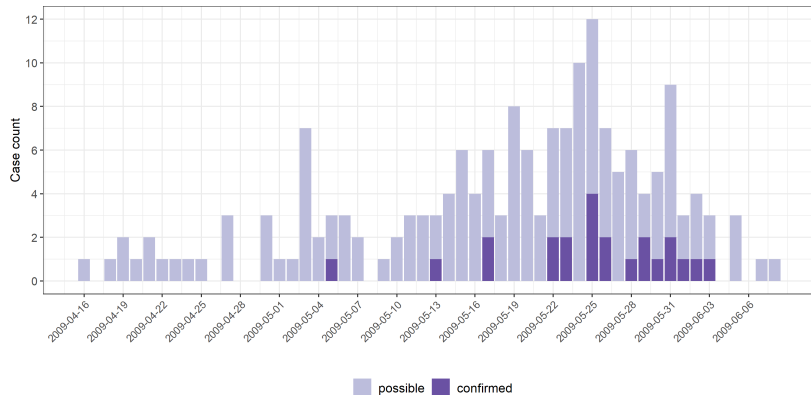
$$\frac{dR(t)}{dt} = \gamma I(t)$$

Computational performance - compared to PMMH

	SEIR, $N = 500$		SEIR, $N = 1000$	
	DA-MCMC	PMMH	DA-MCMC	PMMH
<i>Scenario 1: $r_0 = 1.3, \varphi = 0.2$</i>				
CPU time (minutes)	10.1 (4.5, 19.5)	89.1 (62.9, 114.8)	20.1 (9.5, 29.3)	124.7 (69.6, 287.4)
ESS(log r_0) / CPU time	48.6 (10.4, 134.0)	32.1 (21.9, 49.5)	20.9 (5.0, 55.3)	27.2 (10.3, 43.2)
<i>Scenario 2: $r_0 = 1.3, \varphi = 0.7$</i>				
CPU time (minutes)	13.7 (6.3, 32.3)	118.8 (60.7, 435.4)	21.0 (14.5, 27.2)	137.7 (100.2, 212.4)
ESS(log r_0) / CPU time	88.1 (19.1, 252.1)	24.7 (5.0, 40.4)	33.0 (9.8, 95.2)	15.6 (7.9, 23.5)
<i>Scenario 3: $r_0 = 2.0, \varphi = 0.2$</i>				
CPU time (minutes)	16.9 (8.5, 26.2)	119.5 (93.3, 132.4)	31.0 (22.0, 37.6)	139.8 (114.0, 156.4)
ESS(log r_0) / CPU time	17.2 (6.0, 43.5)	19.0 (10.5, 27.6)	5.7 (3.1, 10.3)	18.9 (13.3, 24.6)
<i>Scenario 4: $r_0 = 2.0, \varphi = 0.7$</i>				
CPU time (minutes)	17.9 (10.9, 22.4)	183.0 (87.0, 255.1)	34.0 (22.1, 51.1)	141.9 (122.1, 157.3)
ESS(log r_0) / CPU time	26.4 (10.1, 57.2)	11.5 (5.6, 22.1)	7.9 (3.4, 15.7)	13.5 (7.7, 18.4)

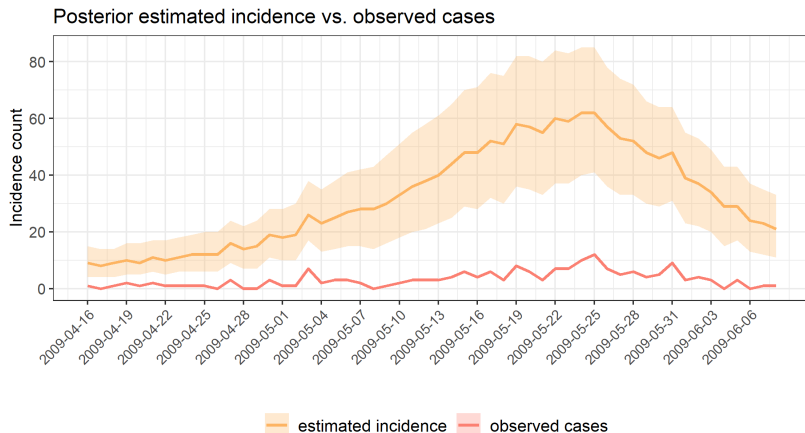
Application to the 2009 H1N1 outbreak

Daily identified possible and confirmed H1N1 cases.



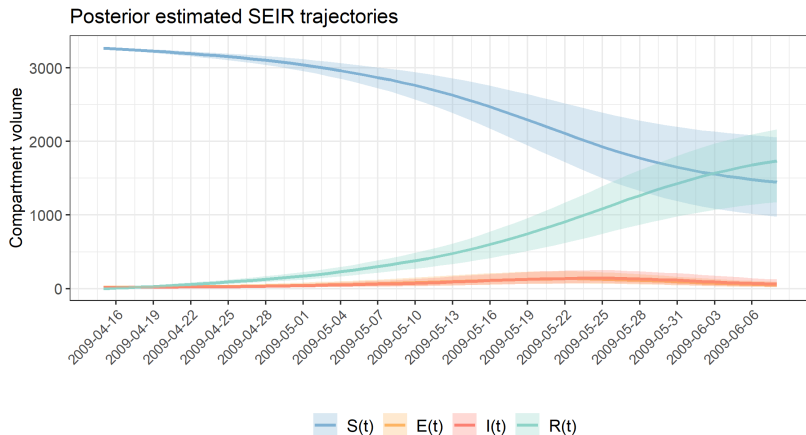
SEIR model fit to H1N1 outbreak - Results

Estimated case detection rate: 0.10 (90% CrI: 0.08 - 0.15).



SEIR model fit to H1N1 outbreak - Results

Estimated reproduction number: 1.55 (90% CrI: 1.29 - 1.85).



Part 8

Volatility models

Stochastic process $\{S(t), t > 0\}$ is used to represent the evolution of an asset value through time.

- share price
- index value (FTSE, Dow-Jones, S&P 500 ...)
- exchange rate
- interest rate

The stochastic process model forms the basis of various financial decision making processes.

Typically use a process driven by Brownian motion $W(t)$, with drift, satisfying the SDE

$$dS(t) = \mu S(t)dt + \sigma S(t)dW(t)$$

We can think of a discrete approximation

$$S_{t+\delta t} - S_t \simeq \mu S_t \delta t + \sigma S_t W_t$$

where $W_t \sim N(0, 1)$, or

$$S_{t+\delta t} | S_t \sim \text{Normal}(S_t(1 + \mu), S_t^2 \sigma^2)$$

σ , the *volatility*, is presumed constant.

- If $Y(t) = \log S(t)$, then

$$dY(t) = \{\mu - \sigma^2/2\}dt + \sigma dW(t).$$

- Typically, the log asset value is observed discretely at intervals of length Δ , say, yielding data $y_0, \dots, y_n$
- Then

$$Y_t - Y_{t-1} \sim \text{Normal}((\mu - \sigma^2/2)\Delta, \sigma^2\Delta)$$

with these increments independent; these are the *log returns*.

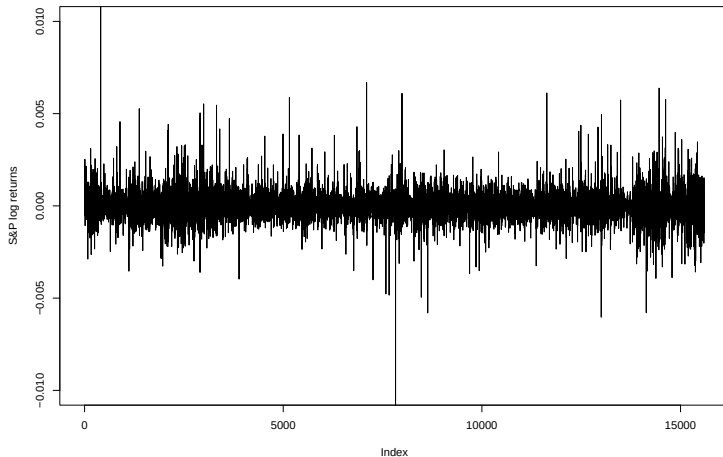
The problem is ... this model does not capture observed behaviour.
... every assumption underlying the Black-Scholes model is rejected by the type of data that are routinely used in practice.

Barndorff-Nielsen & Shephard (2001)

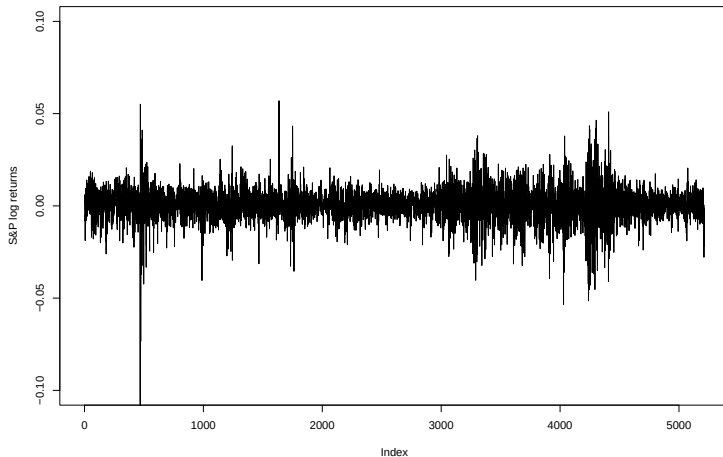
Using y_t^2 as a proxy for the volatility for interval $(\triangle(t-1), \triangle t)$, we observe that

- volatility **is not constant**
- volatility exhibits **autocorrelation**
- marginal distribution of $y_1, \dots, y_n$ is **leptokurtic** (heavy-tailed).

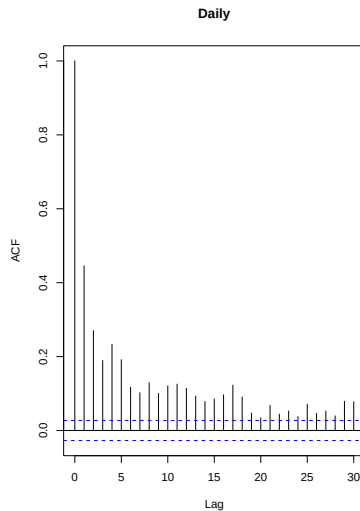
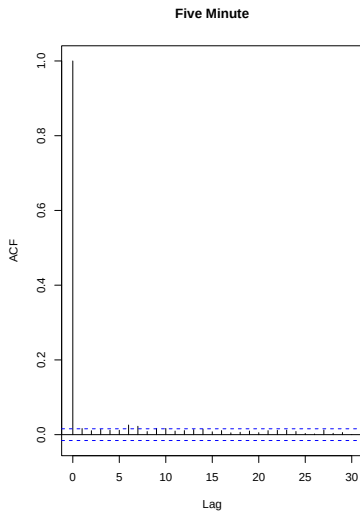
S&P 500 Index (log scale): Five minute



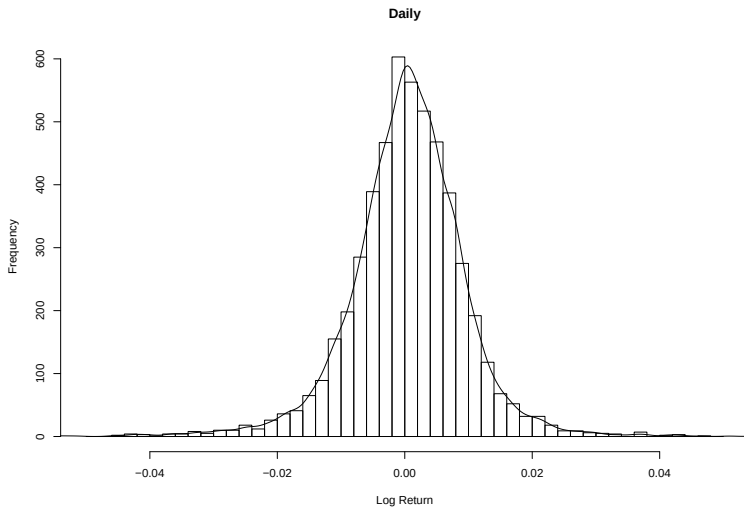
S&P 500 Index (log scale): Daily



S&P 500 Index: ACF for Squared log-returns



S&P 500 Index: Skewness



Allow volatility to vary *stochastically* over time

$$dS(t) = \mu S(t)dt + \sigma(t)S(t)dW(t)$$

i.e. let $\{\sigma^2(t), > 0\}$ follow some stochastic process, or SDE.

- Random volatility increases kurtosis of log returns
- Correlation in $\{\sigma^2(t)\}$ induces correlation in square of log returns

We will model $X(t) \equiv \log \sigma(t)$ as a latent stochastic process; we begin in discrete time.

The simplest *stochastic volatility* (SV) model takes the form

$$Y_t = \exp(X_t/2) V_t$$

$$X_t = \mu_X + \phi(X_{t-1} - \mu_X) + W_t$$

where

- Y_t is the log return at time t
- $V_t \sim \text{Normal}(0, 1)$, and $W_t \sim \text{Normal}(0, \sigma_W^2)$ are IID series.
- $\{X_t\}$ interpreted as the random and uneven flow of new information into the market, and ϕ is the persistence in the volatility.

If we square Y_t and take logarithms, then

$$\log Y_t^2 = X_t + \log V_t^2.$$

If V_t has a standard normal distribution, then $\log V_t^2$ has a log-chi-square distribution with mean $\psi(1) - \log 2 \simeq -1.2704$ and variance $\pi^2/2 \simeq 4.9348$, where $\psi(\cdot)$ is the *digamma* function

$$\psi(z) = \frac{d}{dz} \log \Gamma(z) = \frac{\dot{\Gamma}(z)}{\Gamma(z)}.$$

Thus, if $V_t^* = \log V_t^2 + 1.2704$, then $\{V_t\}$ is IID with mean zero and variance $\pi^2/2$, we may rewrite the model as

$$\log Y_t^2 = -1.2704 + X_t + V_t^*.$$

Hence $\{\log Y_t^2\}$ is a *linear* process which is the sum of

- the AR(1) process $\{X_t\}$
- a white noise process V_t^* .

and $\log Y_t^2$ behaves (approximately) as an ARMA(1, 1) process.

For this process, the Kalman filter and SMC methods work well.

The incomplete data likelihood is

$$\begin{aligned}\mathcal{L}_n(\boldsymbol{\theta}) &= \int p(y_{1:n}, x_{1:n}; \boldsymbol{\theta}) dx_{1:n} \\ &= \int p(y_{1:n} \mid x_{1:n}; \boldsymbol{\theta}) p(x_{1:n}; \boldsymbol{\theta}) dx_{1:n}.\end{aligned}$$

Again, the integral requires the use of numerical procedures and this makes the estimation $\boldsymbol{\theta}$ quite involved.

If $\boldsymbol{\theta} = (\phi, \phi, \sigma_W^2)^\top$, then the model is specified by

- the distribution of the observations conditional on the log volatilities, $p(y_{1:n}|X_{1:n})$,
- the distribution of the log-volatilities conditional on the hyperparameters, $p(X_{1:n}|\boldsymbol{\theta})$
- the prior distribution of the parameters, $\pi_0(\boldsymbol{\theta})$.

The joint posterior distribution is

$$\pi_n(X_{1:n}, \boldsymbol{\theta} | y_{1:n}) \propto p(y_{1:n} | X_{1:n}) p(X_{1:n} | \boldsymbol{\theta}) \pi_0(\boldsymbol{\theta}).$$

Here we can adopt standard MCMC: initialize the values

$$x_{1:n}^{(1)}, \boldsymbol{\theta}^{(1)}$$

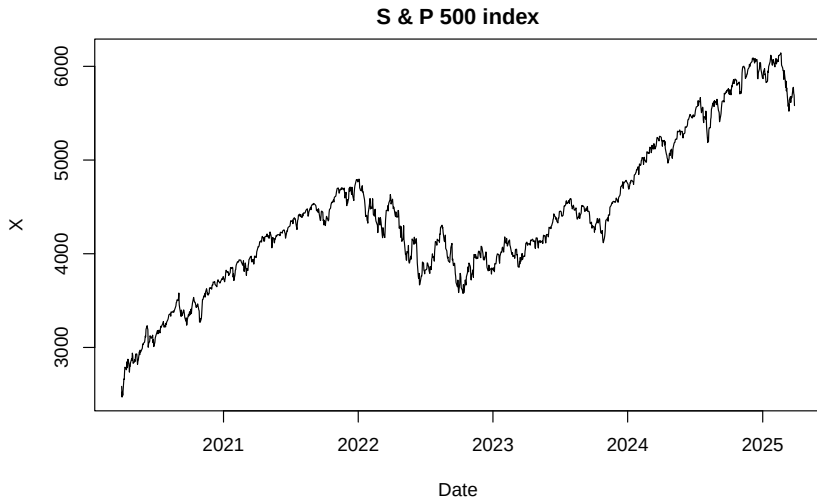
and then for iteration $s = 2, 3, \dots$, sample

- $\boldsymbol{\theta}^{(s)}$ from $\pi_n(\boldsymbol{\theta}|X_{1:n}, y_{1:n})$, assuming $X_{1:n} = x_{1:n}^{(s-1)}$
- $x_{1:n}^{(s)}$ from $\pi_n(X_{1:n}|\boldsymbol{\theta}, y_{1:n})$, assuming $\boldsymbol{\theta} = \boldsymbol{\theta}^{(s)}$

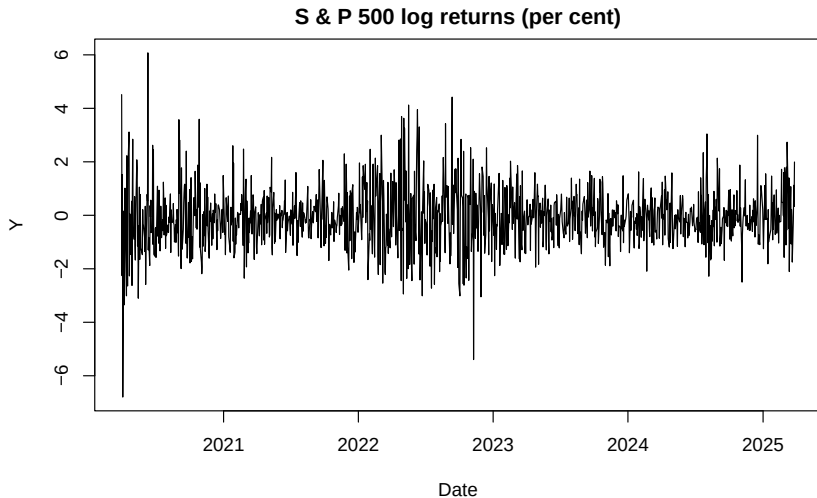
and store the samples $\{\boldsymbol{\theta}^{(s)}, x_{1:n}^{(s)}\}$ over many iterations.

R package `stochvol` provides an efficient MCMC implementation; in the following example $n = 1255$, MCMC runs for 100000 iterations in about 1 min.

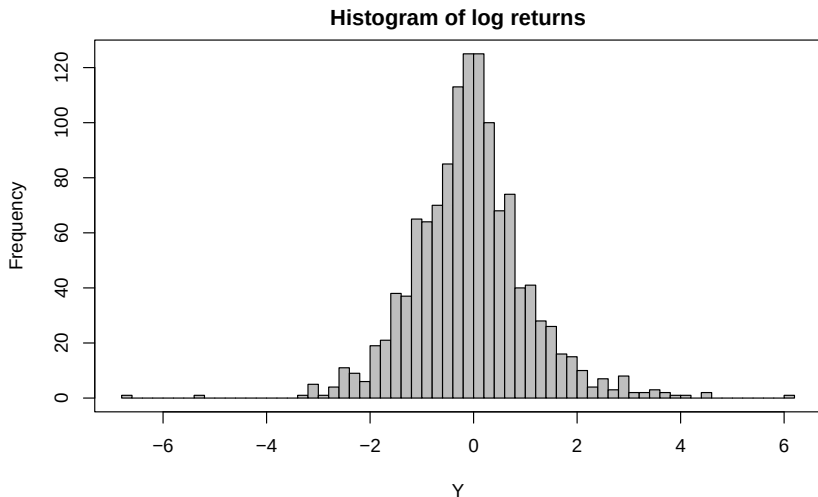
8.4. Bayesian Inference



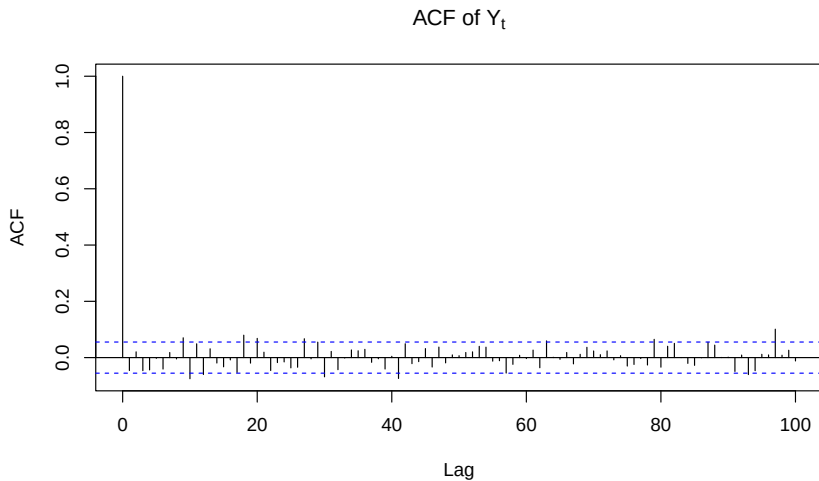
8.4. Bayesian Inference



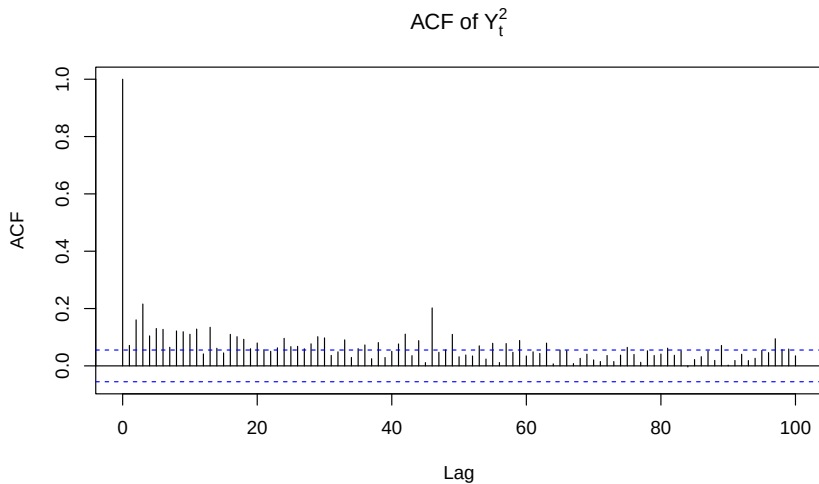
8.4. Bayesian Inference



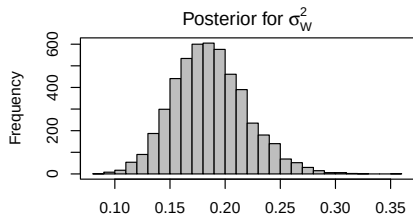
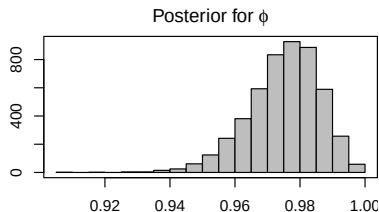
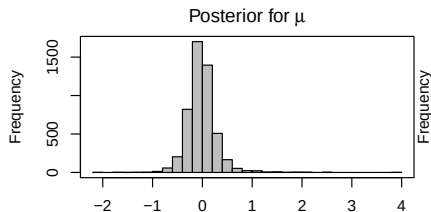
8.4. Bayesian Inference



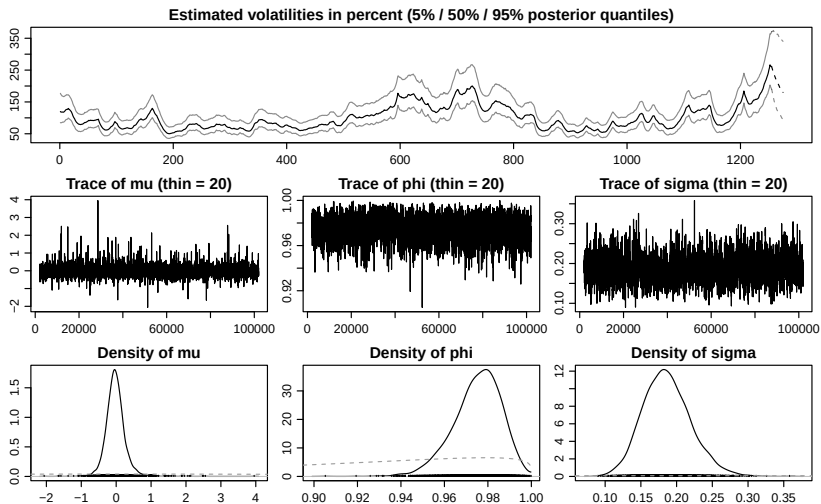
8.4. Bayesian Inference



8.4. Bayesian Inference



8.4. Bayesian Inference



- Student-t outcome model
- Multivariate versions
- Continuous time (diffusion) model for $\{\log \sigma(t)\}$
 - ▶ Heston model
 - ▶ allows for *leverage*

Part 9

Lévy Process models

Barndorff-Nielsen & Shephard (BNS) suggest to make SV process follow the Ornstein-Uhlenbeck (OU) equation

$$d\sigma^2(t) = -\lambda\sigma^2(t) dt + dZ(\lambda t)$$

The process $\sigma^2(t)$ is a stationary process, with autocorrelation function given for $s, t > 0$ by

$$\text{Corr} [\sigma^2(s), \sigma^2(s+t)] = \exp\{-\lambda t\}.$$

$\{Z(t), t > 0\}$ is a *Lévy process* with a scalar *time-change* induced by λ .

Lévy Process, $\{Z(t), t \geq 0\}$:

(i) $Z(0) = 0$.

(ii) *Stochastic continuity*; for all $t, h, \epsilon > 0$,

$$\lim_{h \rightarrow 0} P[|Z_{t+h} - Z_t| \geq \epsilon] = 0.$$

(iii) *Independent increments*: random quantities

$$Z_{t_0}, Z_{t_1} - Z_{t_0}, Z_{t_3} - Z_{t_2}, \dots, Z_{t_n} - Z_{t_{n-1}}$$

are independent.

(iv) *Stationary increments*; for $t, h \geq 0$, the distribution of $Z_{t+h} - Z_t$ does not depend on t .

The Brownian motion driving the Black-Scholes SDE *observation* equation is a Lévy Process; the BNS formulation allows for more general driving processes in the *volatility* equation.

For modelling real data, we must choose the form of the driving process $Z(t)$. We wish to capture the observed behaviour of the log-returns

- autocorrelation (volatility clustering)
- leptokurtosis (heavy-tailed log returns distribution)

A distribution, F with density $f(x)$, characteristic function ϕ_X and characteristic exponent $\Psi(s)$,

$$\Psi(s) = -\log \mathbb{E}[\exp\{isX\}] = -\log \phi_X(s)$$

is *infinitely divisible* if, for each $n \in \mathbb{Z}^+$

$$\phi_X(s) = \{\phi_n(s)\}^n$$

where ϕ_n is the characteristic function of some other distribution.

Furthermore, F is infinitely divisible if and only if there exists

- $a \in \mathbb{R}$,
- a positive semi-definite quadratic form B on $\mathbb{R}$
- a measure $U(dx)$ on $\mathbb{R} \setminus \{0\}$ with density $u(x)$.

such that $\forall s \in \mathbb{R}$

$$\Psi(s) = ias + B(s)/2 + \int_{\mathbb{R}} \left[1 - e^{isx} + isx \mathbf{1}_{\{|x| < 1\}} \right] u(x) dx$$

and

$$\int_{-\infty}^{\infty} \min\{1, x^2\} u(x) dx < \infty.$$

This is the *Lévy-Khintchine formula*, which provides a 1-1 correspondence between infinitely divisible distributions and Lévy processes

- $U(dx)$ is the *Lévy measure* of f
- $u(x)$ is the *Lévy density* of f
- $B(s)$ is the *Gaussian coefficient*.
- $f(x)$ is the *marginal law* of Z .

If $B(s) \equiv 0$, then $\{Z(t)\}$ is termed a *pure-jumps* process.

Example: The Gamma Process

Suppose, for $x > 0$, $B(s) \equiv 0$ and

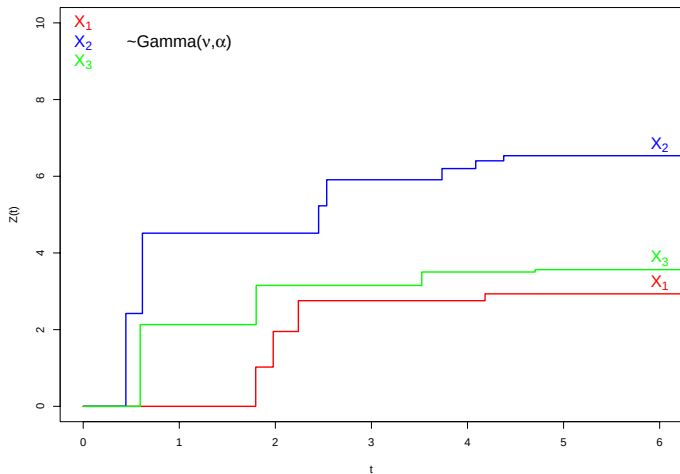
$$u(x) = x^{-1} \nu \alpha e^{-\alpha x}$$

$\{Z(t)\}$ is a *Gamma process*.

- The marginal law is a $\text{Gamma}(\nu, \alpha)$ distribution; the total increment in a finite interval is a $\text{Gamma}(\nu, \alpha)$ random variable.
- In any finite interval, the process has an *infinite* number of jumps; we can simulate sample paths by ordering the jumps in descending order.

The Gamma Process

$\nu = 5, \alpha = 1$:



If

$$d\sigma^2(t) = -\lambda\sigma^2(t)dt + dZ(\lambda t)$$

then we can either

- specify the marginal law of $\sigma^2(t)$ and deduce the nature of $Z(t)$, or
- specify $Z(t)$ and deduce the marginal law of $\sigma^2(t)$

The properties of $Z(t)$ are transmitted to $\sigma^2(t)$; in particular, the marginal law of Z induces the marginal law of $\sigma^2(t)$.

We interpret the $dZ(t)$ term as the (random) change in Z at instant t . If $Z(t)$ is a pure-jumps process, we need *positive* jumps only.

In the BNS OU-SV model:

- Z is termed the *Background Driving Lévy process* (BDLP)
- Z is a pure jumps process, and is non-decreasing
- Z has an associated marginal law (given by the Lévy-Khintchine theorem) that *does not depend on λ* .
- Parameter λ and the time-change control the nature and rate of jumps of $\sigma^2(t)$.

We should pick Z to induce appropriate marginal law for $\sigma^2(t)$.

No theoretical reason to prefer one model over another - should make assessment statistically.

Lévy Measures for $\sigma^2(t)$ and $Z(t)$

- $u(x)$ the Lévy density for the (ID) law of $\sigma^2(t)$
- $w(x)$ the Lévy density for the marginal law of $Z(1)$.
- Under the BNS OU-SV specification

$$w(x) = -u(x) - x \frac{du(x)}{dx}$$

- Tail Mass function (TMF)

$$W^+(x) = \int_x^\infty w(y) dy = xu(x)$$

- Inverse Tail mass function (ITMF)

$$W^-(x) = \inf \{y > 0 : W^+(y) \leq x\}$$

Example:

If $\sigma^2(t)$ has a $\text{Gamma}(\nu, \alpha)$ marginal law, then for $x > 0$,

$$u(x) = x^{-1} \nu e^{-\alpha x}$$

$$w(x) = \alpha \nu e^{-\alpha x}$$

$$W^+(x) = \nu e^{-\alpha x}$$

$$W^-(x) = \max\{0, -\log(x/\nu)/\alpha\}$$

that is, $u(x)$ is the Lévy density of a Gamma process; $\{\sigma^2(t)\}$ is not itself a Gamma process.

Solution to OU equation can be written via a stochastic integral wrt Z

$$\sigma^2(t) = e^{-\lambda t} \sigma^2(0) + e^{-\lambda t} \int_0^t e^{\lambda s} dZ(\lambda s)$$

The Ferguson & Klass *infinite series* representation of $Z(t)$ facilitates

- simulating $Z(t)$
- performing inference about $\sigma^2(t)$

We use the Inverse-Lévy Method: other simulation methods exist.

We have

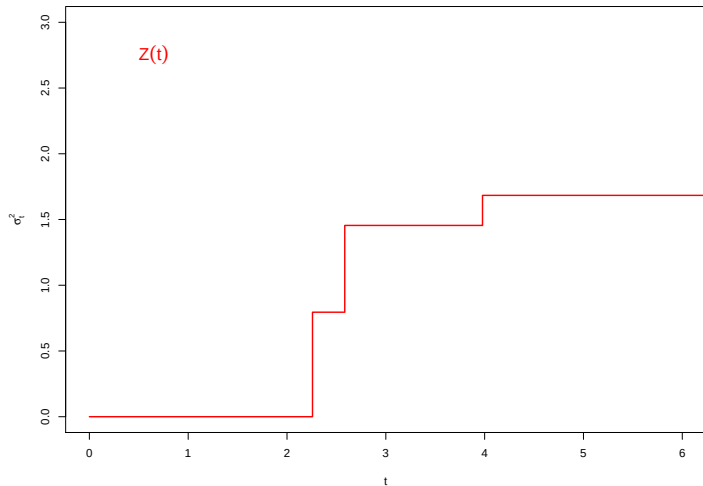
$$\int_0^\Delta h(s) dZ(s) \stackrel{\mathcal{L}}{=} \sum_{j=1}^{\infty} h(\Delta R_j) W^-(A_j/\Delta)$$

where

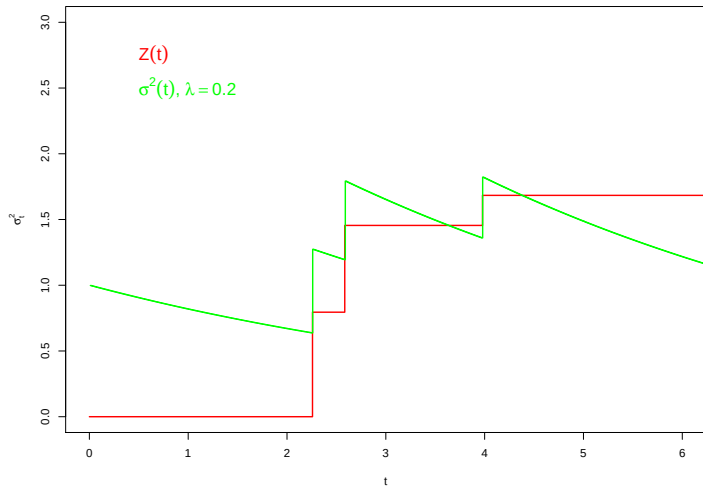
- $A_1 < A_2 < \dots$ are a sequence of event times in a standard Poisson process
- R_j are independent Uniforms
- W^- is the ITMF defined earlier.

Auxiliaries $(a_j, r_j), j = 1, 2, \dots$, facilitate implementation.

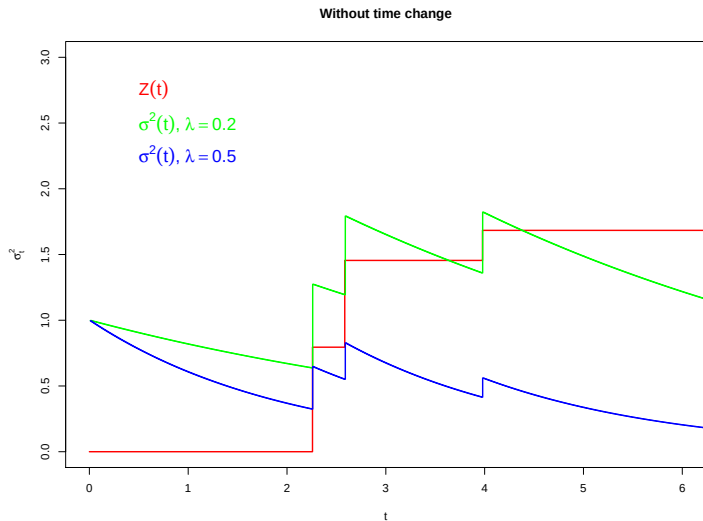
Examples: Realizations of $\{\sigma^2(t)\}$



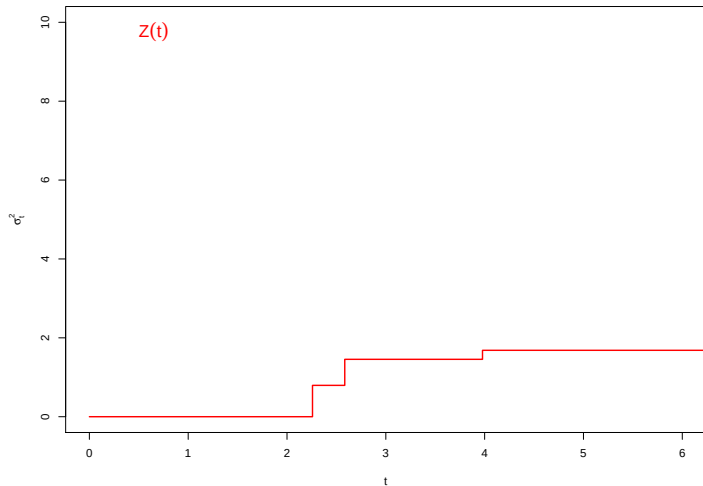
Examples: Realizations of $\{\sigma^2(t)\}$



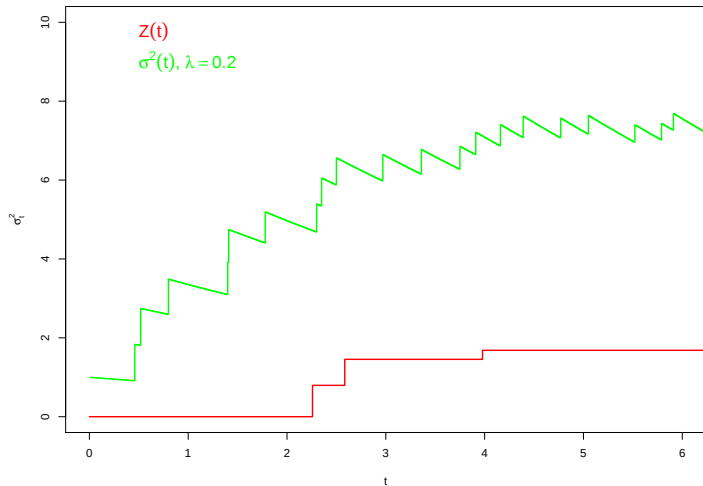
Examples: Realizations of $\{\sigma^2(t)\}$



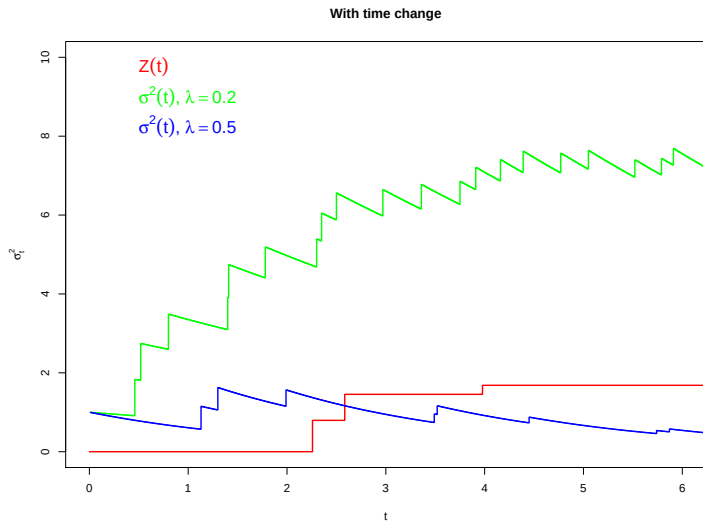
Examples: Realizations of $\{\sigma^2(t)\}$



Examples: Realizations of $\{\sigma^2(t)\}$



Examples: Realizations of $\{\sigma^2(t)\}$



For the likelihood, similarly to the constant volatility case

$$Y_t \sim \text{Normal}((\mu - \sigma_t^2/2)\Delta, \sigma_t^2\Delta)$$

with $Y_1, \dots, Y_n$ independent, where

$$\sigma_t^2 = \int_0^{\Delta t} d\sigma^2(s) - \int_0^{\Delta(t-1)} d\sigma^2(s)$$

is the *discretely observed* volatility.

In the BNS OU-SV models, we have the following parameters to make inference about

- λ , autocorrelation decay parameter.
- θ , the parameters of the marginal law of $\sigma^2(t)$.
- the list of latent variables that form the (Ferguson & Klass) representation of the BDLP, that is

$$(a_1, a_2, \dots) \quad (r_1, r_2, \dots).$$

- $\sigma^2(0)$, initial value of volatility process.
- More sophisticated model components can be added.

Useful speedups :

- Joint updates for auxiliaries and parameters
- Targeted Proposals : target updates of auxiliaries (a_j, r_j) in areas of poor fit to data under current state. Interval t
 - ▶ has return y_t
 - ▶ has collection of auxiliaries $\boldsymbol{\theta}^{(t)} = (\mathbf{a}^{(t)}, \mathbf{r}^{(t)})$
 - ▶ has a discretely observed volatility $\sigma_t^2 \equiv \sigma_t^2(\boldsymbol{\theta}^{(t)})$ given any set of parameter values
 - ▶ we target update of $\boldsymbol{\theta}^{(t)}$ where residual $(y_t - \mu_t)/\sigma_t$ is largest, that is, select t with probability proportional to

$$\exp\{(y_t - \mu_t)^2 / \sigma_t^2\}$$

and correct in Metropolis-Hastings ratio.

Most interest has focussed on the *Generalized Inverse Gaussian* (GIG) family of distributions as suitable as marginal laws for $\sigma^2(t)$.

MCMC inference has been implemented for a *Gamma* marginal law; the ITMF takes a particularly simple form, and the infinite series in the series representation of Z reduces to a *finite* one.

This simplifies the computation significantly, although the MCMC is still not trivial.

The problem is ... this model may not capture observed behaviour.

- no particular theoretical support for a Gamma marginal
- autocorrelation decay is not always exponential
- volatility process is perhaps non-stationary

Need extension to full GIG marginal model: for parameters $\gamma \in \mathbb{R}$ and $\nu, \alpha > 0$, the $GIG(\gamma, \nu, \alpha)$ pdf takes the form

$$f(x) = \frac{(\alpha/\nu)^\gamma}{2K_\gamma(\nu\alpha)} x^{\gamma-1} \exp \left\{ -\frac{1}{2} (\nu^2 x^{-1} + \alpha^2 x) \right\}$$

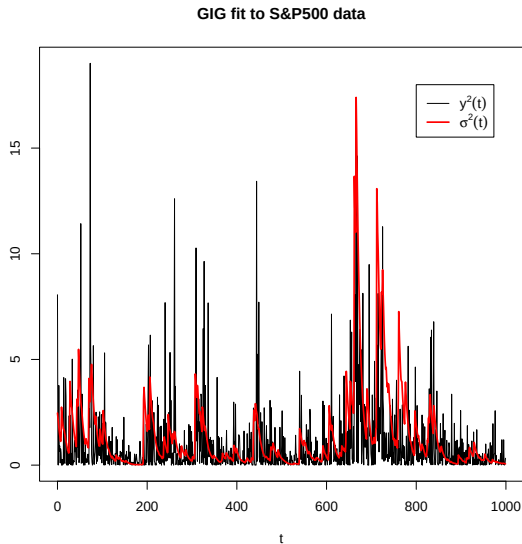
for $x > 0$, where K_ν is a modified Bessel function of the third kind.

GIG special cases:

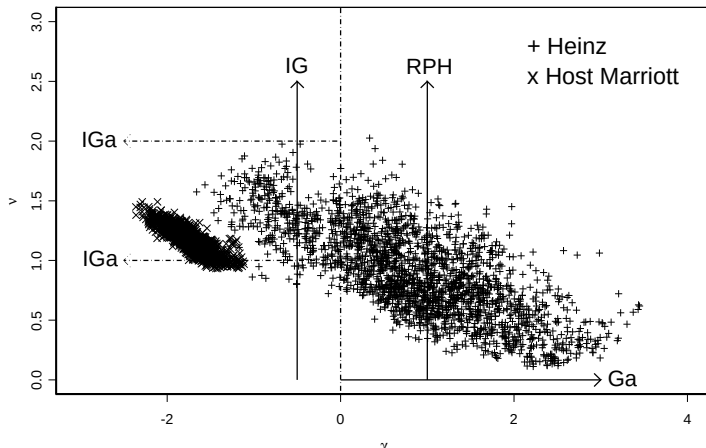
Form of $GIG(\gamma, \nu, \alpha)$ distribution	Distribution
$GIG(\nu, 0, \sqrt{2\alpha})$	$Ga(\nu, \alpha)$
$GIG(1, \nu, \alpha)$	$PH(\nu, \alpha)$
$GIG(-\nu, \sqrt{2\alpha}, 0)$	$IGa(\nu, \alpha)$
$GIG(-\frac{1}{2}, \nu, \alpha)$	$IG(\nu, \alpha)$

Ga is *Gamma*, PH is *positive hyperbolic*, IGa is *Inverse Gamma*, IG is *Inverse Gaussian*

GIG Fit to S & P 500 Data



Posterior samples in the (γ, ν) plane for two data sets.



The exponential decay in the autocorrelation provided by the BNS OU-SV model is limited.

- May use *superposition*: set

$$\sigma^2(t) = \sum_{k=1}^K w_k \sigma_k^2(t)$$

where $\sigma_k^2(t)$ is BNS OU with parameter λ_k .

Extension: More General Correlation Structure

- Can also construct volatility processes with richer autocorrelation
e.g. adopt the *moving average model*

$$\sigma^2(t) = \int_{-\infty}^t h(t-s) dZ(s)$$

where $h(t-s) \geq 0$ if $t > s$. We have autocorrelation function

$$\rho(t) = \text{Corr}[\sigma^2(t_0), \sigma^2(t_0 + t)] = \frac{\int_0^\infty h(|t| + s) h(s) ds}{\int_0^\infty h^2(s) ds}.$$

- ▶ Power Decay model

$$h(x) = \frac{1}{(1 + \lambda |x|)^\beta} \quad \beta > 1/2$$

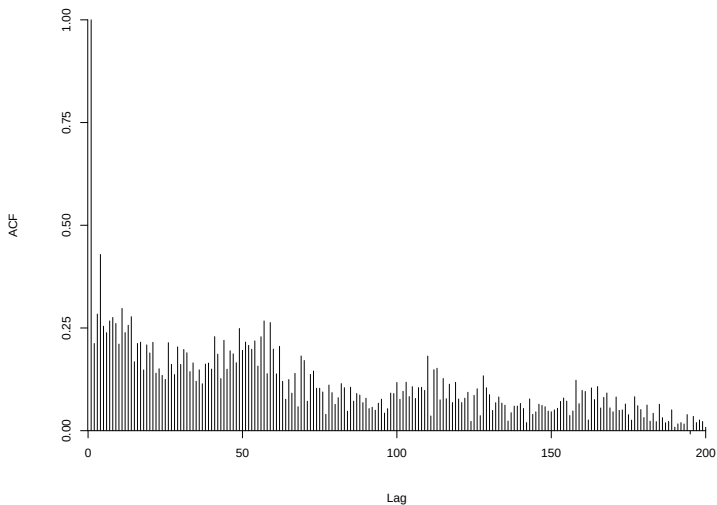
For large lags t , yields an ACF that decays like $t^{-\beta}$. If $1/2 < \beta < 1$, then model exhibits long memory.

- ▶ Fractional OU Lévy Model

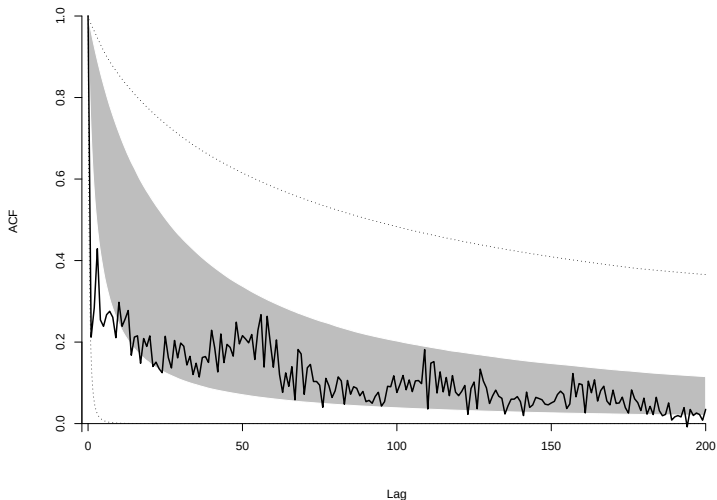
$$h(x) = \sqrt{2\lambda} \frac{\lambda^{\kappa-1}}{\Gamma(\kappa)} x^{\kappa-1} e^{-\lambda x} \quad x \geq 0$$

which has finite variance and short memory if $\kappa > 1/2$.

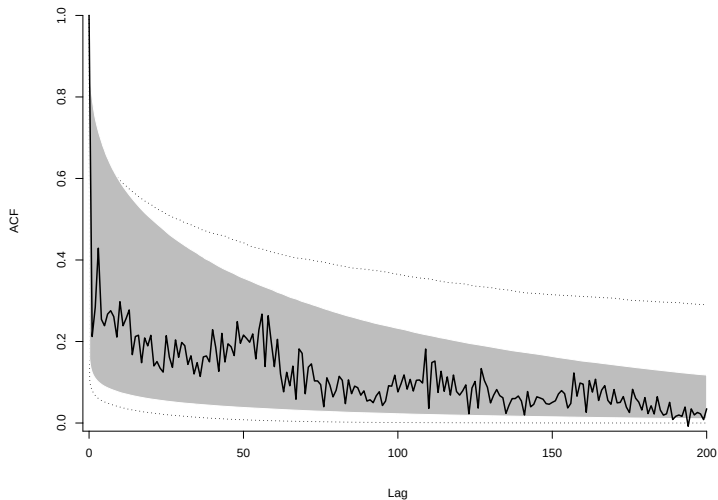
Empirical ACF



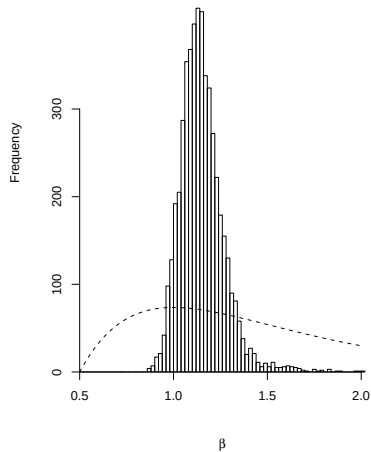
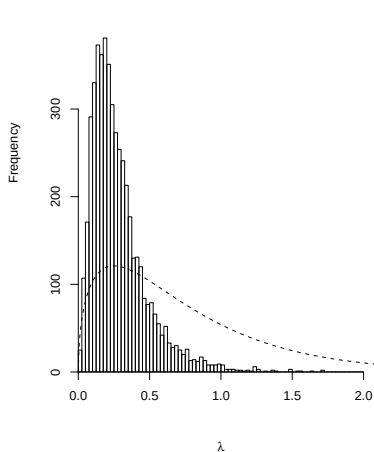
Pointwise 95 % credible interval for ACF: Power Decay Model



Pointwise 95 % credible interval for ACF: fOUL Model



Posterior samples for λ and β : Power Decay Model



Changing the Observation Equation

Motivation: To use a more general observation model to better capture observed market behaviour.

- Some evidence that there is jump-like behaviour in both observation and volatility.
- can be achieved by using a Lévy process with jump components rather than a Brownian motion to drive the *observation* equation.
- Variance-Gamma model (time-changed Brownian motion)

- The BNS-SV models extend in different directions
- Going beyond the Gamma marginal is useful
- Going beyond exponential decay in correlation is vital
- Jump components to driving processes probably needed
- SMC also methods offer attractive realtime analysis

1. Bayesian inference implemented via *augmentation*
2. Ordinary MCMC feasible for most Gaussian models using standard approaches
 - forward/backward *filtering* calculations useful for likelihood calculation
3. SMC and PMCMC developments also useful
4. For Markov latent models inference is feasible