Bayesian calculus

Marie-Pierre Etienne

https://github.com/MarieEtienne

Novembre 2018

Outline

Outline

1. Our goal today
2. Analytical posterior determination
3. Sampling from posterior distribution
4. Importance sampling algorithm
5. Monte Carlo Markov Chain algorithm (MCMC)

Getting the posterior distribution

A posteriori distribution is defined by

$$[\theta|y] = \frac{[y|\theta][\theta]}{[y]}$$

with $[y] = \int_\theta [y|\theta][\theta]d\theta$.

$[y]$ is mostly unavailable.

But nothing can stop us !!

$$[\theta|y] = ?$$

Getting the posterior distribution

A posteriori distribution is defined by

$$[\theta|y] = \frac{[y|\theta][\theta]}{[y]}$$

with $[y] = \int_\theta [y|\theta][\theta] d\theta$.

$[y]$ is mostly unavailable.

But nothing can stop us !!

$$[\theta|y] = ?$$

Getting the posterior distribution

A posteriori distribution is defined by

$$[\theta|y] = \frac{[y|\theta][\theta]}{[y]}$$

with $[y] = \int_\theta [y|\theta][\theta] d\theta$.

$[y]$ is mostly unavailable.

But nothing can stop us !!

$$[\theta|y] = ?$$

Getting the posterior distribution

A posteriori distribution is defined by

$$[\theta|y] = \frac{[y|\theta][\theta]}{[y]}$$

with $[y] = \int_\theta [y|\theta][\theta] d\theta$.

$[y]$ is mostly unavailable.

But nothing can stop us !!

$$[\theta|y] = ?$$

Outline

Binomial example

- Data model :

$$Y \sim \mathcal{B}(n, p), \quad n \text{ known}$$

- Prior uniform

$$p \sim \mathcal{U}(0, 1)$$

-

$$[p|y] = ?$$

Normal example

- Model : $Y_k = \beta_0 + \beta_1 x_k + E_k, \quad E_k \overset{ind}{\sim} \mathcal{N}(0, \sigma^2)$
- Normal prior on $\theta = (\beta_0, \beta_1)$, ($\sigma^2$ assumed to be known)

$$[\beta_0, \beta_1] = \mathcal{N}(\mu_{prior}, \Lambda_{prior}),$$

  with Λ_{prior} denoting the precision matrix.
- Posterior distribution

$$[\beta_0, \beta_1 | y] \sim \mathcal{N}(\mu_{post}, \Lambda_{post})$$

  with

$$\Lambda_{post} = \left(\frac{\mathbf{X}^{\mathrm{T}}\mathbf{X}}{\sigma^2} + \boldsymbol{\Lambda}_{prior} \right)$$

$$\mu_{post} = \left(\frac{\mathbf{X}^{\mathrm{T}}\mathbf{X}}{\sigma^2} + \boldsymbol{\Lambda}_{prior} \right)^{-1} \left(\frac{\mathbf{X}^{\mathrm{T}}\mathbf{Y}}{\sigma^2} + \boldsymbol{\Lambda}_{prior}\boldsymbol{\mu}_{prior} \right)$$

Normal example

- Model : $Y_k = \beta_0 + \beta_1 x_k + E_k, \quad E_k \overset{ind}{\sim} \mathcal{N}(0, \sigma^2)$
- Normal prior on $\theta = (\beta_0, \beta_1)$, ($\sigma^2$ assumed to be known)

$$[\beta_0, \beta_1] = \mathcal{N}(\mu_{prior}, \Lambda_{prior}),$$

with Λ_{prior} denoting the precision matrix.

- Posterior distribution

$$[\beta_0, \beta_1 | y] \sim \mathcal{N}(\mu_{post}, \Lambda_{post})$$

with

$$\Lambda_{post} = \left(\frac{\mathbf{X}^{\mathrm{T}}\mathbf{X}}{\sigma^2} + \mathbf{\Lambda}_{prior} \right)$$

$$\mu_{post} = \left(\frac{\mathbf{X}^{\mathrm{T}}\mathbf{X}}{\sigma^2} + \mathbf{\Lambda}_{prior} \right)^{-1} \left(\frac{\mathbf{X}^{\mathrm{T}}\mathbf{Y}}{\sigma^2} + \mathbf{\Lambda}_{prior} \boldsymbol{\mu}_{prior} \right)$$

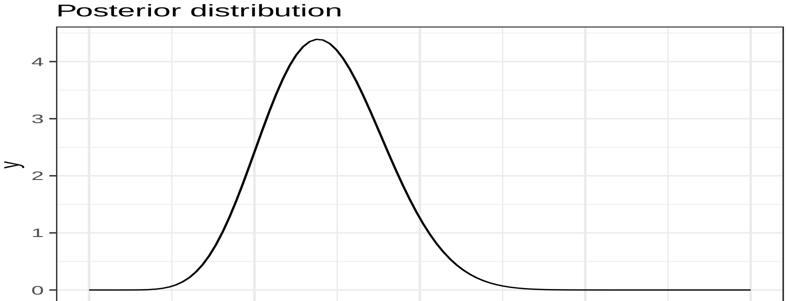
Outline

Why a sample is mostly enough ?

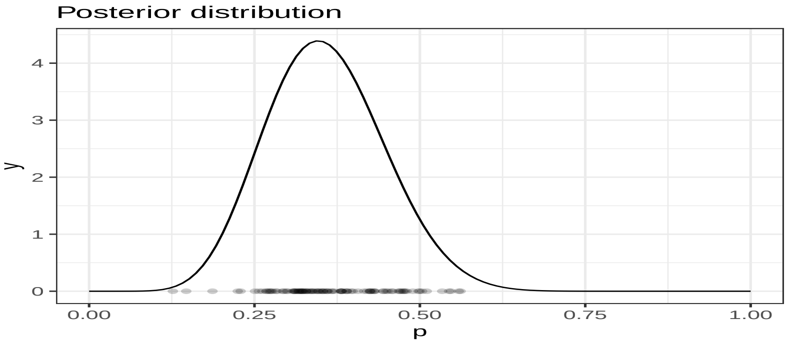
```
xseq <- seq(0, 1, length.out=100)
sh1  <- 10
sh2  <- 18
density.post <- dbeta(xseq, shape1 = sh1, shape2 = sh2)
df <- data.frame(x=xseq, y =density.post)
p <- ggplot(data=df, aes(x=x, y=y)) +geom_line() + xlab('p') + ggtitle('Posterior distribution'
suppressMessages(ggsave(filename = 'figMC1.pdf', width = 5, height = 4))
```

```
n1 <- 100
sim <- rbeta(n = n1, shape1 = sh1, shape2 = sh2)
p.MC <- p + geom_point(data= data.frame(x=sim, y=rep(0,n1)), aes(x=x,y=y), alpha=0.2 )
print(p.MC)
```

```
suppressMessages(ggsave(filename = 'figMC2.pdf', width = 5, height = 4))
```



Why a sample is mostly enough ?

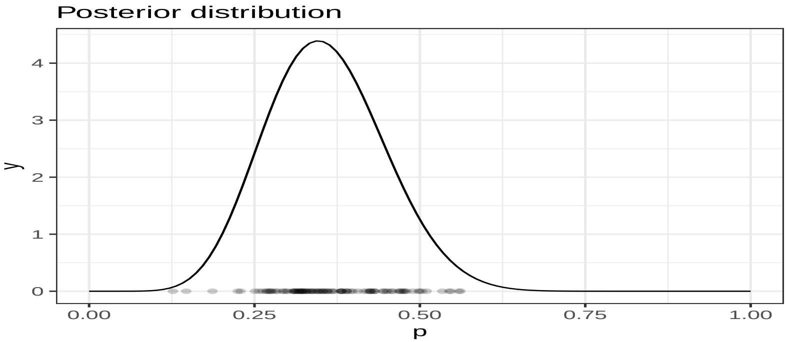


- $E[p|y] \approx ?$
- $CI_{0.95}(p) \approx ?$

```
 df <- data.frame(c('Mean', 'CIInf', 'CISup'), 'theory'=c(sh1/(sh1+sh2), qbeta(0.05, shape1 = s
n2 <- 1000
sim <- rbeta(n = n2, shape1 = sh1, shape2 = sh2)

df =cbind(df,c(mean(sim), quantile(sim, probs = 0.05), quantile(sim, probs = 0.95)))
names(df) = c('Sum','Theory', paste0('MC',n1), paste0('MC',n2) )
print(df)
```

Why a sample is mostly enough ?



- $E[p|y] \approx ?$
- $CI_{0.95}(p) \approx ?$

```
 df <- data.frame(c('Mean', 'CIInf', 'CISup'), 'theory'=c(sh1/(sh1+sh2), qbeta(0.05, shape1 = s
n2 <- 1000
sim <- rbeta(n = n2, shape1 = sh1, shape2 = sh2)

df =cbind(df,c(mean(sim), quantile(sim, probs = 0.05), quantile(sim, probs = 0.95)))
names(df) = c('Sum','Theory', paste0('MC',n1), paste0('MC',n2) )
print(df)
```

Outline

Importance sampling approach

Main idea :

$$
\begin{aligned}
E_{d_X}({\color{blue}h(X)}) &= \int_u {\color{blue}h(u)}\, d_X(u) du = \int_u h(u)\, \frac{d_X(u)}{{\color{green}d_Z(u)}} {\color{green}d_Z(u)} du \\
&= \int_u {\color{red}h(u)\, \frac{d_X(u)}{d_Z(u)}} d_Z(u) du = E_{d_Z}\left({\color{red}h(Z)\frac{d_X(Z)}{d_Z(Z)}}\right)
\end{aligned}
$$

```
xseq <- seq(0, 1, length.out=100)
sh1  <- 10
sh2  <- 18
density.post <- dbeta(xseq, shape1 = sh1, shape2 = sh2)
df <- data.frame(x=xseq, y =density.post)

p <- ggplot(data=df, aes(x=x, y=y)) +geom_line() + xlab('p') + ggtitle('Posterior distribution'
print(p)
```

```
proposal <- rnorm(n1, mean=0.5, sd=0.5)

p1 <- p + geom_point(data=data.frame(x=proposal, y=rep(0,n1)),  col='red', alpha=0.2)
print(p1)
```

```
suppressMessages(ggsave(filename = 'IS1.pdf', width = 5, height = 4))

weight <- dbeta(proposal, shape1 = sh1, shape2 = sh2)/dnorm(proposal, mean=0.5, sd=0.5)

p2 <- p + geom_point(data=data.frame(x=proposal, y=rep(0,n1), weight=weight), aes(x=x, y=y, si
```

Importance sampling approach

Main idea :

$$
\begin{aligned}
E_{d_X}(h(X)) &= \int_u h(u)\, d_X(u) du = \int_u h(u)\, \frac{d_X(u)}{d_Z(u)} d_Z(u) du \\
&= \int_u h(u)\, \frac{d_X(u)}{d_Z(u)} d_Z(u) du = E_{d_Z}\left(h(Z) \frac{d_X(Z)}{d_Z(Z)} \right)
\end{aligned}
$$

```
xseq <- seq(0, 1, length.out=100)
sh1  <- 10
sh2  <- 18
density.post <- dbeta(xseq, shape1 = sh1, shape2 = sh2)
df <- data.frame(x=xseq, y =density.post)

p <- ggplot(data=df, aes(x=x, y=y)) +geom_line() + xlab('p') + ggtitle('Posterior distribution'
print(p)
```

```
proposal <- rnorm(n1, mean=0.5, sd=0.5)

p1 <- p + geom_point(data=data.frame(x=proposal, y=rep(0,n1)),  col='red', alpha=0.2)
print(p1)
```

```
suppressMessages(ggsave(filename = 'IS1.pdf', width = 5, height = 4))

weight <- dbeta(proposal, shape1 = sh1, shape2 = sh2)/dnorm(proposal, mean=0.5, sd=0.5)

p2 <- p + geom_point(data=data.frame(x=proposal, y=rep(0,n1), weight=weight), aes(x=x, y=y, si
```

Importance sampling approach

Main idea :

$$E_{d_X}(h(X)) = \int_u h(u)\, d_X(u) du = \int_u h(u)\, \frac{d_X(u)}{d_Z(u)} d_Z(u) du$$
$$= \int_u h(u)\, \frac{d_X(u)}{d_Z(u)} d_Z(u) du = E_{d_Z}\left(h(Z) \frac{d_X(Z)}{d_Z(Z)} \right)$$

```
xseq <- seq(0, 1, length.out=100)
sh1  <- 10
sh2  <- 18
density.post <- dbeta(xseq, shape1 = sh1, shape2 = sh2)
df <- data.frame(x=xseq, y =density.post)

p <- ggplot(data=df, aes(x=x, y=y)) +geom_line() + xlab('p') + ggtitle('Posterior distribution'
print(p)
```

```
proposal <- rnorm(n1, mean=0.5, sd=0.5)

p1 <- p + geom_point(data=data.frame(x=proposal, y=rep(0,n1)),  col='red', alpha=0.2)
print(p1)
```

```
suppressMessages(ggsave(filename = 'IS1.pdf', width = 5, height = 4))

weight <- dbeta(proposal, shape1 = sh1, shape2 = sh2)/dnorm(proposal, mean=0.5, sd=0.5)

p2 <- p + geom_point(data=data.frame(x=proposal, y=rep(0,n1), weight=weight), aes(x=x, y=y, si
```

Importance sampling approach

Main idea :

$$E_{d_X}(h(X)) = \int_u h(u)\, d_X(u) du = \int_u h(u)\, \frac{d_X(u)}{d_Z(u)} d_Z(u) du$$
$$= \int_u h(u)\, \frac{d_X(u)}{d_Z(u)} d_Z(u) du = E_{d_Z}\left(h(Z) \frac{d_X(Z)}{d_Z(Z)} \right)$$

```
xseq <- seq(0, 1, length.out=100)
sh1  <- 10
sh2  <- 18
density.post <- dbeta(xseq, shape1 = sh1, shape2 = sh2)
df <- data.frame(x=xseq, y =density.post)

p <- ggplot(data=df, aes(x=x, y=y)) +geom_line() + xlab('p') + ggtitle('Posterior distribution'
print(p)
```

```
proposal <- rnorm(n1, mean=0.5, sd=0.5)

p1 <- p + geom_point(data=data.frame(x=proposal, y=rep(0,n1)),  col='red', alpha=0.2)
print(p1)
```

```
suppressMessages(ggsave(filename = 'IS1.pdf', width = 5, height = 4))

weight <- dbeta(proposal, shape1 = sh1, shape2 = sh2)/dnorm(proposal, mean=0.5, sd=0.5)

p2 <- p + geom_point(data=data.frame(x=proposal, y=rep(0,n1), weight=weight), aes(x=x, y=y, si
```

Importance sampling approach

Main idea :

$$E_{d_X}(h(X)) = \int_u h(u)\, d_X(u)du = \int_u h(u)\, \frac{d_X(u)}{d_Z(u)} d_Z(u)du$$
$$= \int_u h(u)\, \frac{d_X(u)}{d_Z(u)} d_Z(u)du = E_{d_Z}\left(h(Z)\frac{d_X(Z)}{d_Z(Z)}\right)$$

```
xseq <- seq(0, 1, length.out=100)
sh1  <- 10
sh2  <- 18
density.post <- dbeta(xseq, shape1 = sh1, shape2 = sh2)
df <- data.frame(x=xseq, y =density.post)

p <- ggplot(data=df, aes(x=x, y=y)) +geom_line() + xlab('p') + ggtitle('Posterior distribution'
print(p)
```

```
proposal <- rnorm(n1, mean=0.5, sd=0.5)

p1 <- p + geom_point(data=data.frame(x=proposal, y=rep(0,n1)),  col='red', alpha=0.2)
print(p1)
```

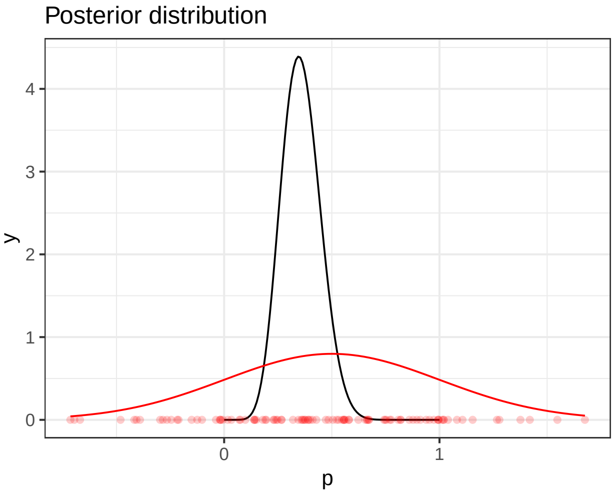
```
suppressMessages(ggsave(filename = 'IS1.pdf', width = 5, height = 4))

weight <- dbeta(proposal, shape1 = sh1, shape2 = sh2)/dnorm(proposal, mean=0.5, sd=0.5)

p2 <- p + geom_point(data=data.frame(x=proposal, y=rep(0,n1), weight=weight), aes(x=x, y=y, si
```

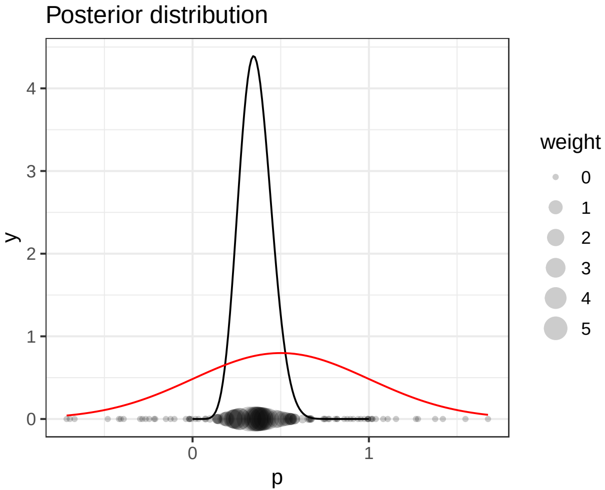
IS algorithm : graphical point of view

1. Step 1 : sample from proposal $(N = 100)$



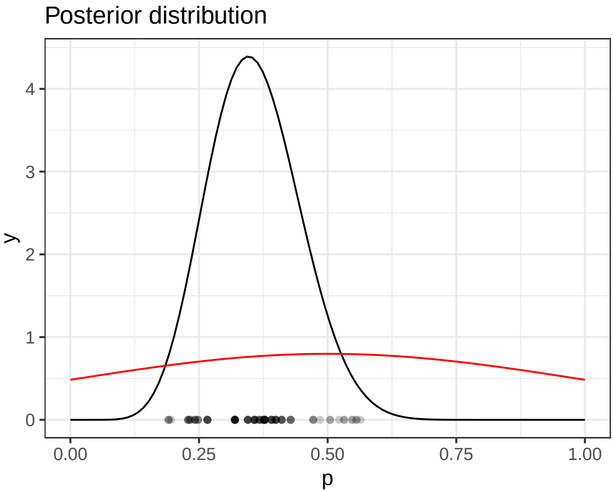
IS algorithm : graphical point of view

2. Step 2 : compute weight ($N = 100$)



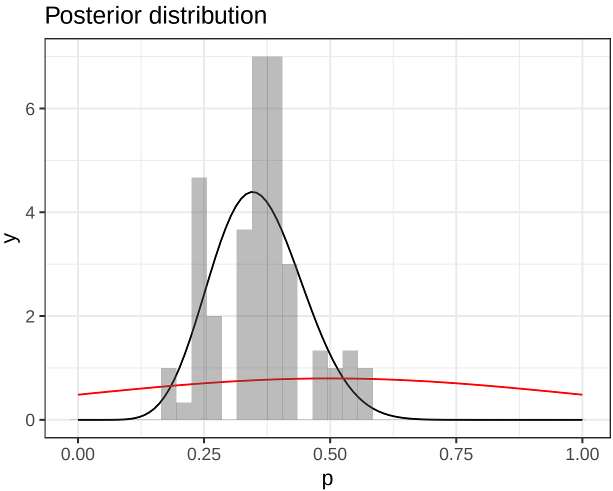
IS algorithm : graphical point of view

❸ Step 3 : Resample to get unweighted sample $(N = 100)$



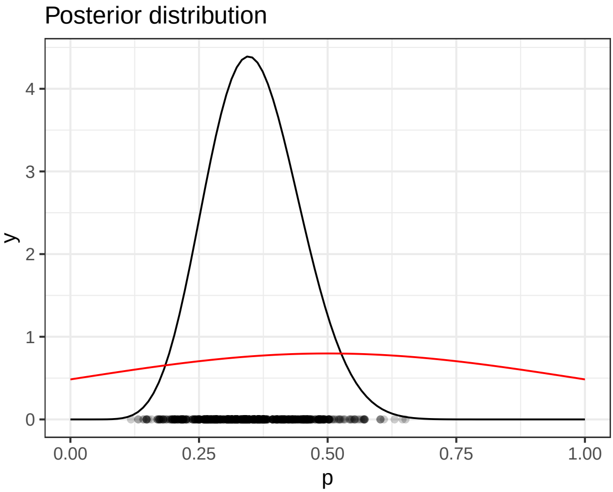
IS algorithm : graphical point of view

③ Step 3 : Resample to get unweighted sample $(N = 100)$



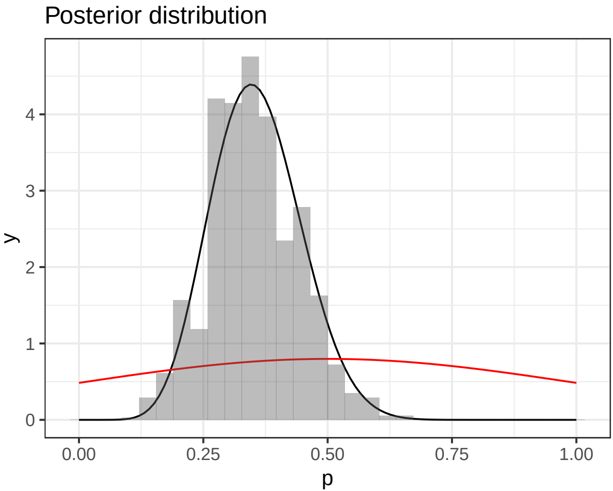
IS algorithm : graphical point of view

❸ Step 3 : Resample to get unweighted sample $(N = 1000)$



IS algorithm : graphical point of view

3. Step 3 : Resample to get unweighted sample ($N = 1000$)



Outline

1. Our goal today
2. Analytical posterior determination
3. Sampling from posterior distribution
4. Importance sampling algorithm
5. Monte Carlo Markov Chain algorithm (MCMC)

Markov chain definition

A Markov chain is a sequence of random variables $X_1, \ldots, X_n)$ verifying the Markov property.

$$[X_{i+1}|X_{1:i}] = [X_{i+1}|X_i].$$

Markov chain definition

A Markov chain is a sequence of random variables $X_1, \ldots, X_n)$ verifying the Markov property.

$$[X_{i+1}|X_{1:i}] = [X_{i+1}|X_i].$$

Markov chain example

Random walk

$$X_{i+1} = X_i + E_{i+1}, \quad E_{i+1} \overset{ind}{\sim} \mathcal{U}(\{-1, 1\})$$

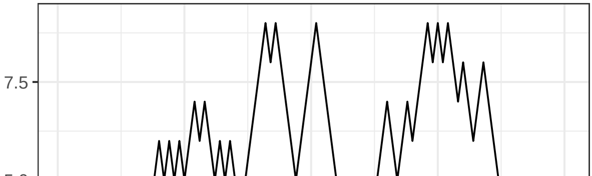
(X_i) is a Markov chain.

```
n <- 100
E <- sample(c(-1,1), replace=TRUE, size= n)
X <- cumsum(E)
p1 <- ggplot(data=data.frame(time=seq(1,n), X=X)) + geom_line(aes(x=time, y=X))
print(p1)
```

```
suppressMessages(ggsave(filename = 'RW1.pdf', width = 5, height = 4))
```

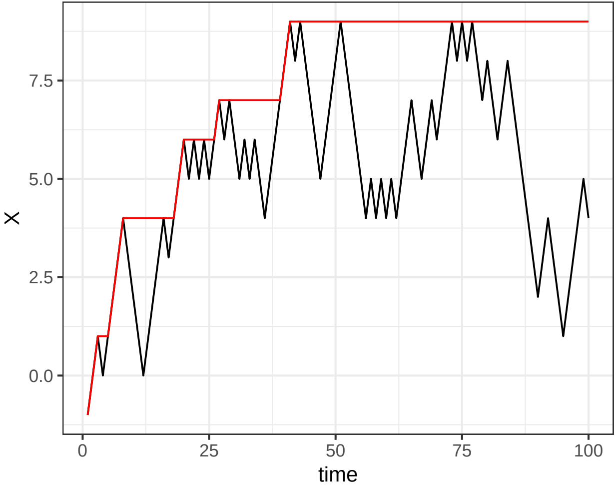
```
Z <- sapply(1:n, function(i_){max(X[1:i_])})
p2<-  p1 +geom_line(data = data.frame(time=seq(1,n), Z=Z), aes(x=time, y=Z), col='red')
print(p2)
```

```
suppressMessages(ggsave(filename = 'SuppRW1.pdf', width = 5, height = 4))
```



Markov chain example

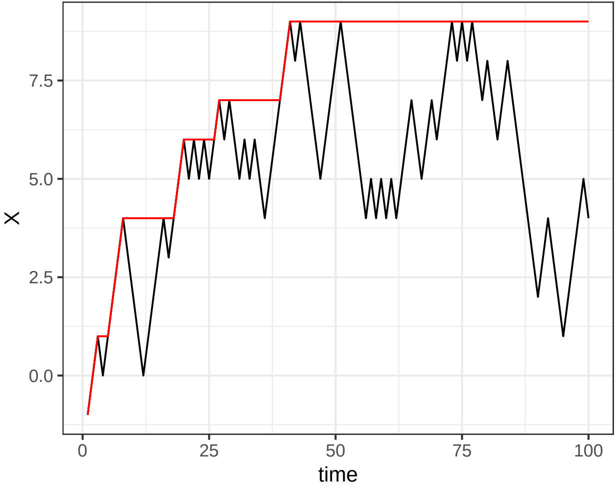
Supremum of a random walk $Z_i = max_{k=1}^{i} max(X_k)$,



(Z_i) is not a Markov chain.

Markov chain example

Supremum of a random walk $Z_i = max_{k=1}^{i} max(X_k)$,



(Z_i) is not a Markov chain.

Markov chain properties

Definition : ν is a stationnay distribution if and only if

$$X_i \sim \nu \Longrightarrow X_{i+1} \sim \nu$$

Example :

$$X_1 \sim \mathcal{B}(p_{init}), \quad X_{i+1}|X_i \sim \mathcal{B}(p_{X_i})$$

```
n     <- 100
pinit <- 1/3
pr    <- c(0.2, 0.6)
X     <- rep(NA,n)

X[1] <- sample(c(0,1), size=1, prob = c(1-pinit, pinit))
for( i in 1:(n-1)){
 X[i+1] <-   sample(x=c(0,1), size = 1,
                    prob = c(1-pr[X[i]+1], pr[X[i]+1 ]))
}

p <- ggplot(data=data.frame(time = seq(1,n), X=X), aes(x=time, y=X)) + geom_line()
suppressMessages(ggsave(plot = p , filename = 'OnOff.pdf', width = 5, height = 4))
```

Markov chain properties

Definition : ν is a stationnay distribution if and only if

$$X_i \sim \nu \Longrightarrow X_{i+1} \sim \nu$$

Example :

$$X_1 \sim \mathcal{B}(p_{init}), \quad X_{i+1}|X_i \sim \mathcal{B}(p_{X_i})$$

```
n     <- 100
pinit <- 1/3
pr    <- c(0.2, 0.6)
X     <- rep(NA,n)

X[1] <- sample(c(0,1), size=1, prob = c(1-pinit, pinit))
for( i in 1:(n-1)){
 X[i+1] <-   sample(x=c(0,1), size = 1,
                    prob = c(1-pr[X[i]+1], pr[X[i]+1 ]))
}
```

```
p <- ggplot(data=data.frame(time = seq(1,n), X=X), aes(x=time, y=X)) + geom_line()
suppressMessages(ggsave(plot = p , filename = 'OnOff.pdf', width = 5, height = 4))
```

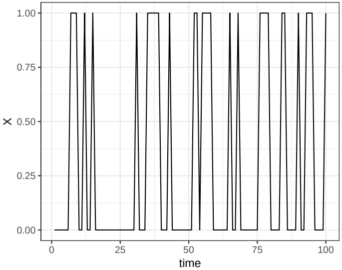
Markov chain properties

Definition : ν is a stationnay distribution if and only if

$$X_i \sim \nu \Longrightarrow X_{i+1} \sim \nu$$

Example :

$$X_1 \sim \mathcal{B}(p_{init}), \quad X_{i+1}|X_i \sim \mathcal{B}(p_{X_i})$$



Distribution of X_1, X_2, ... ?

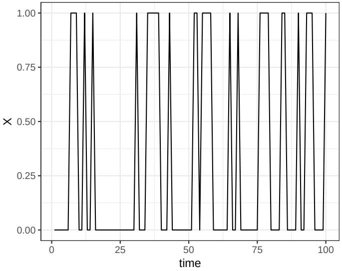
Markov chain properties

Definition : ν is a stationnay distribution if and only if

$$X_i \sim \nu \Longrightarrow X_{i+1} \sim \nu$$

Example :

$$X_1 \sim \mathcal{B}(p_{init}), \quad X_{i+1}|X_i \sim \mathcal{B}(p_{X_i})$$



Distribution of X_1, X_2, ... ?

Markov chain properties

Ergodic property :

If a Markov chain (X_i) is irreducible, aperiodic and recurrent then there is exists a unique stationnary distribution π and

$$[X_n] \underset{n\to\infty}{\longrightarrow} \pi.$$

If a Markov chain (X_i) is reversible $\left([X_i][X_{i+1}|X_i] = [X_{i+1}][X_i|X_{i+1}]\right)$ then this markov chain has a stationnary distribution.

Consequences of the ergodic theorem

If (X_n) is a Markov chain with stationnary distribution, for any initial distribu
$[X_1]$, $[X_n]$ is close to the stationnary distribution.

Back to the example : stationnary distribution is $\pi = (0.7, 0.3)$

```
freq = table(X)/n
print(freq)
```

```
## X
##    0    1
## 0.69 0.31
```

Consequences of the ergodic theorem

If (X_n) is a Markov chain with stationnary distribution, for any initial distribu
$[X_1]$, $[X_n]$ is close to the stationnary distribution.

Back to the example : stationnary distribution is $\pi = (0.7, 0.3)$

```
freq = table(X)/n
print(freq)
```

```
## X
##    0    1
## 0.69 0.31
```

Metropolis Hastings algorithm

Key idea : building a reversible Markov chain with $[\theta|y]$ as stationnary distribution

1. Initialization $\theta^{(0)}$ an admissible initial value
2. For i in 1:nIter

- Propose a new candidate value $\theta_c^{(i)}$ sampled from a proposal distribution $g(.|\theta^{(i-1)})$
- Compute Metropolis Hastings ratio

$$r_i = \frac{[y|\theta_c^{(i)}][\theta_c^{(i)}]}{[y|\theta^{(i-1)}][\theta^{(i-1)}]} \frac{g(\theta^{(i-1)}|\theta^{(i)})}{g(\theta_c^{(i)}|\theta^{(i-1)})}$$

  - Define

$$\theta^{(i)} = \begin{cases} \theta_c^{(i)} \text{ with probablity } min(r_i, 1) \\ \theta_c^{(i-1)} \text{ with probablity } 1 - min(r_i, 1) \end{cases}$$

Metropolis Hastings algorithm

Key idea : building a reversible Markov chain with $[\theta|y]$ as stationnary distribution

1. Initialization $\theta^{(0)}$ an admissible initial value
2. For i in 1:nIter

- Propose a new candidate value $\theta_c^{(i)}$ sampled from a proposal distribution $g(.|\theta^{(i-1)})$
- Compute Metropolis Hastings ratio

$$r_i = \frac{[y|\theta_c^{(i)}][\theta_c^{(i)}]}{[y|\theta^{(i-1)}][\theta^{(i-1)}]} \frac{g(\theta^{(i-1)}|\theta^{(i)})}{g(\theta_c^{(i)}|\theta^{(i-1)})}$$

  - Define

$$\theta^{(i)} = \begin{cases} \theta_c^{(i)} \text{ with probablity } min(r_i, 1) \\ \theta_c^{(i-1)} \text{ with probablity } 1 - min(r_i, 1) \end{cases}$$