

# AdaptBUGS?

Christophe Andrieu

<mailto:c.andrieu@bris.ac.uk>



February 12, 2006

# Contents

<b>1</b>	<b>Some background</b>	<b>3</b>
<b>2</b>	<b>Controlled MCMC</b>	<b>9</b>
<b>3</b>	<b>Theoretical perspective</b>	<b>21</b>
<b>4</b>	<b>Conclusions</b>	<b>29</b>

# 1 Some background

## 1.1 The Metropolis-Hastings (MH) update

- Most, if not all MCMC algorithms rely on the Metropolis-Hastings (MH) update to sample from a distribution  $\pi$ .

# 1 Some background

## 1.1 The Metropolis-Hastings (MH) update

- Most, if not all MCMC algorithms rely on the Metropolis-Hastings (MH) update to sample from a distribution  $\pi$ .
- Even when a Gibbs sampler is possible, a MH update might be preferable.

# 1 Some background

## 1.1 The Metropolis-Hastings (MH) update

- Most, if not all MCMC algorithms rely on the Metropolis-Hastings (MH) update to sample from a distribution  $\pi$ .
- Even when a Gibbs sampler is possible, a MH update might be preferable.
- However it requires the choice of a family of proposal distribution  $q(x, \cdot)$  for  $x \in X$ .

# 1 Some background

## 1.1 The Metropolis-Hastings (MH) update

- Most, if not all MCMC algorithms rely on the Metropolis-Hastings (MH) update to sample from a distribution  $\pi$ .
- Even when a Gibbs sampler is possible, a MH update might be preferable.
- However it requires the choice of a family of proposal distribution  $q(x, \cdot)$  for  $x \in X$ .
- This is both a strength and a weakness.

# 1 Some background

## 1.1 The Metropolis-Hastings (MH) update

- Most, if not all MCMC algorithms rely on the Metropolis-Hastings (MH) update to sample from a distribution  $\pi$ .
- Even when a Gibbs sampler is possible, a MH update might be preferable.
- However it requires the choice of a family of proposal distribution  $q(x, \cdot)$  for  $x \in X$ .
- This is both a strength and a weakness.

## 1.2 The update

- The MH update proceeds as follows :

## 1.2 The update

- The MH update proceeds as follows :
- At iteration  $i + 1$ , given  $X_i = x$ :

## 1.2 The update

- The MH update proceeds as follows :
- At iteration  $i + 1$ , given  $X_i = x$ :
  1. Propose a transition  $y \sim q(x, \cdot)$ .

## 1.2 The update

- The MH update proceeds as follows :
- At iteration  $i + 1$ , given  $X_i = x$ :
  1. Propose a transition  $y \sim q(x, \cdot)$ .
  2. Calculate the acceptance probability

$$\alpha(x, y) = 1 \wedge \frac{\pi(y)q(y, x)}{\pi(x)q(x, y)}.$$

## 1.2 The update

- The MH update proceeds as follows :
- At iteration  $i + 1$ , given  $X_i = x$ :
  1. Propose a transition  $y \sim q(x, \cdot)$ .
  2. Calculate the acceptance probability

$$\alpha(x, y) = 1 \wedge \frac{\pi(y)q(y, x)}{\pi(x)q(x, y)}.$$

- a)  $X_{i+1} = y$  with probability  $\alpha(x, y)$

## 1.2 The update

- The MH update proceeds as follows :
- At iteration  $i + 1$ , given  $X_i = x$ :
  1. Propose a transition  $y \sim q(x, \cdot)$ .
  2. Calculate the acceptance probability

$$\alpha(x, y) = 1 \wedge \frac{\pi(y)q(y, x)}{\pi(x)q(x, y)}.$$

- a)  $X_{i+1} = y$  with probability  $\alpha(x, y)$
- b) Otherwise,  $X_{i+1} = x$ .

## 1.3 Example and ergodicity issues

- The choice of  $q$  is key to the success of the MCMC approach.

## 1.3 Example and ergodicity issues

- The choice of  $q$  is key to the success of the MCMC approach.
- For example if

$$q_{\theta}(x, y) = \frac{1}{\sqrt{2\pi\theta^2}} \exp\left(\frac{-1}{2\theta^2} (y - x)^2\right).$$

## 1.3 Example and ergodicity issues

- The choice of  $q$  is key to the success of the MCMC approach.
- For example if

$$q_{\theta}(x, y) = \frac{1}{\sqrt{2\pi\theta^2}} \exp\left(\frac{-1}{2\theta^2} (y - x)^2\right).$$

the variance of

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

## 1.3 Example and ergodicity issues

- The choice of  $q$  is key to the success of the MCMC approach.
- For example if

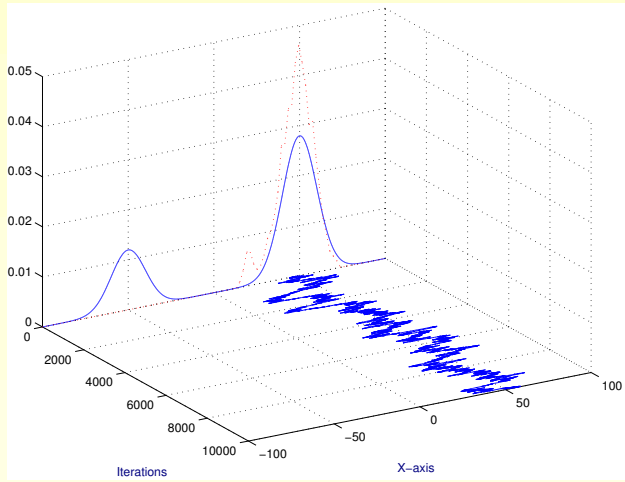
$$q_{\theta}(x, y) = \frac{1}{\sqrt{2\pi\theta^2}} \exp\left(\frac{-1}{2\theta^2} (y - x)^2\right).$$

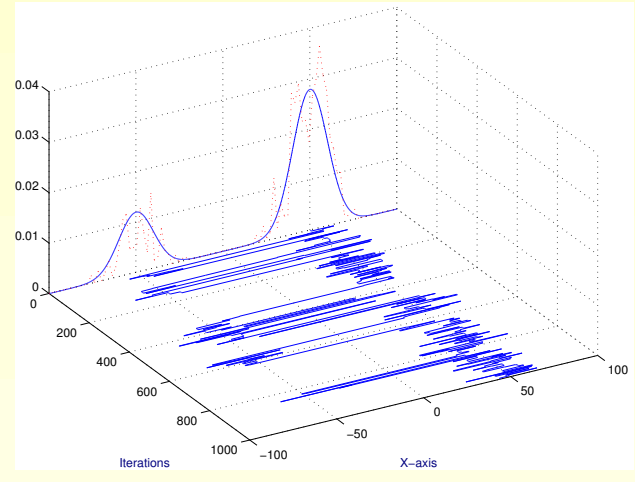
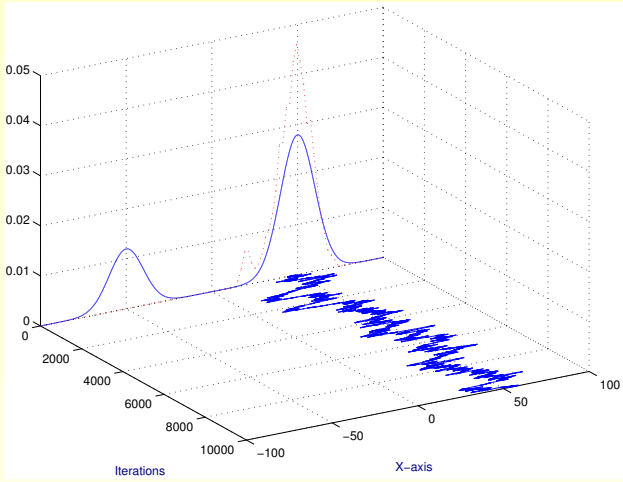
the variance of

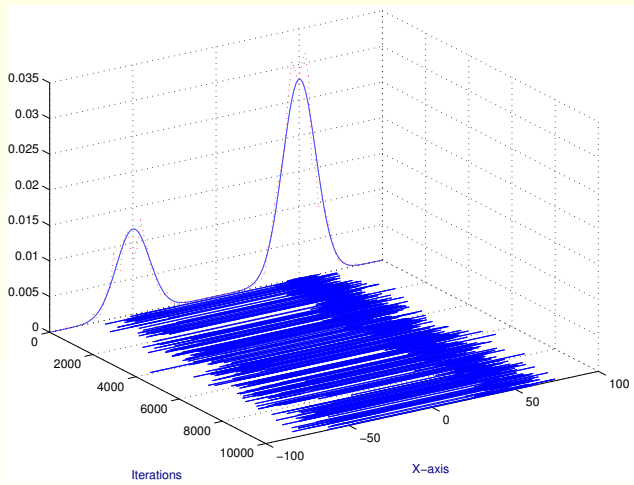
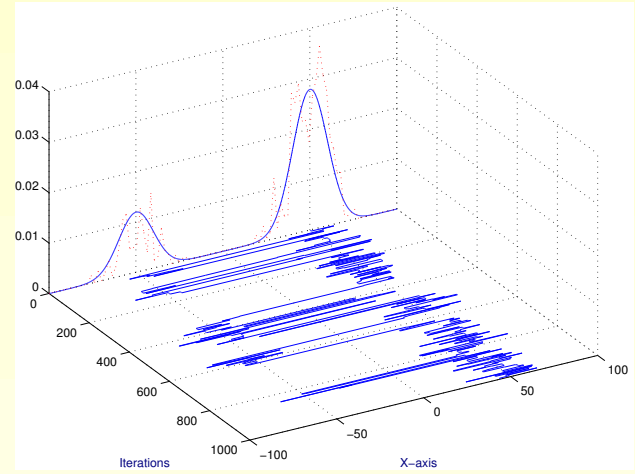
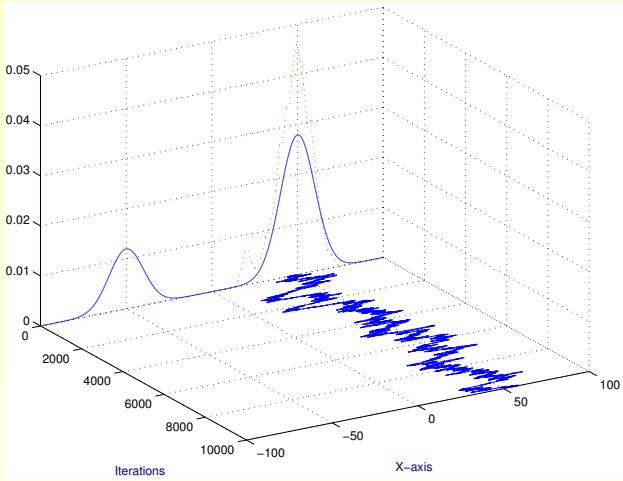
$$\hat{I}_N(f) = \frac{1}{N} \sum_{i=1}^N f(X_i).$$

is large for values of  $\theta^2$  that are either too small or too large.









## 1.4 A more general setup

- More generally, many algorithms can be described as mixtures of MH updates  $\{K_i\}$ .

## 1.4 A more general setup

- More generally, many algorithms can be described as mixtures of MH updates  $\{K_i\}$ .
- For example, the transition kernel  $P_\theta$  of the Markov chain is

$$P_\theta(x, dy) = \sum_{i=1}^n w_i(\theta) K_i(x, dy; \theta),$$

## 1.4 A more general setup

- More generally, many algorithms can be described as mixtures of MH updates  $\{K_i\}$ .
- For example, the transition kernel  $P_\theta$  of the Markov chain is

$$P_\theta(x, dy) = \sum_{i=1}^n w_i(\theta) K_i(x, dy; \theta),$$

where for any  $\theta \in \Theta$ ,  $\{w_i(\theta)\}_i$  is a probability distribution.

## 1.5 Tuning of $\theta$

In practice  $\theta$  is adapted by hand, using trial runs, with more or less well defined criteria.

## 1.5 Tuning of $\theta$

In practice  $\theta$  is adapted by hand, using trial runs, with more or less well defined criteria.

This often requires time and a certain degree of expertise.

## 1.5 Tuning of $\theta$

In practice  $\theta$  is adapted by hand, using trial runs, with more or less well defined criteria.

This often requires time and a certain degree of expertise.

This suggests the following questions:

## 1.5 Tuning of $\theta$

In practice  $\theta$  is adapted by hand, using trial runs, with more or less well defined criteria.

This often requires time and a certain degree of expertise.

This suggests the following questions:

1. What do we want to optimize?

## 1.5 Tuning of $\theta$

In practice  $\theta$  is adapted by hand, using trial runs, with more or less well defined criteria.

This often requires time and a certain degree of expertise.

This suggests the following questions:

1. What do we want to optimize?

## 1.5 Tuning of $\theta$

In practice  $\theta$  is adapted by hand, using trial runs, with more or less well defined criteria.

This often requires time and a certain degree of expertise.

This suggests the following questions:

1. What do we want to optimize?
2. Can the tuning be done in an **automatic** way,

## 1.5 Tuning of $\theta$

In practice  $\theta$  is adapted by hand, using trial runs, with more or less well defined criteria.

This often requires time and a certain degree of expertise.

This suggests the following questions:

1. What do we want to optimize?
2. Can the tuning be done in an **automatic** way, therefore easing their use by non-specialists

## 1.5 Tuning of $\theta$

In practice  $\theta$  is adapted by hand, using trial runs, with more or less well defined criteria.

This often requires time and a certain degree of expertise.

This suggests the following questions:

1. What do we want to optimize?
2. Can the tuning be done in an **automatic** way, therefore easing their use by non-specialists (and “specialists” ...)

## 1.5 Tuning of $\theta$

In practice  $\theta$  is adapted by hand, using trial runs, with more or less well defined criteria.

This often requires time and a certain degree of expertise.

This suggests the following questions:

1. What do we want to optimize?
2. Can the tuning be done in an **automatic** way, therefore easing their use by non-specialists (and “specialists” ...)
3. A dream solution would consist of tuning  $\theta$  while producing samples from  $\pi$ .

## 2 Controlled MCMC

## 2 Controlled MCMC

### 2.1 The literature

- Work on adaptation is not new.

## 2 Controlled MCMC

### 2.1 The literature

- Work on adaptation is not new.
- Reviews on the subject of adaptation can be found for example in [Tierney&Mira 1999], [Andrieu&Robert 2001], [Frigessi 2002].

## 2 Controlled MCMC

### 2.1 The literature

- Work on adaptation is not new.
- Reviews on the subject of adaptation can be found for example in [Tierney&Mira 1999], [Andrieu&Robert 2001], [Frigessi 2002].
- We focus here on vanishing adaptation.

## 2.2 The Finnish example: the symmetric random walk Metropolis (SRWM) algorithm

- We consider the Metropolis algorithm, here in a multivariate context.

## 2.2 The Finnish example: the symmetric random walk Metropolis (SRWM) algorithm

- We consider the Metropolis algorithm, here in a multivariate context.
- The proposal distribution is  $\mathcal{N}(x, \Gamma)$ .

## 2.2 The Finnish example: the symmetric random walk Metropolis (SRWM) algorithm

- We consider the Metropolis algorithm, here in a multivariate context.
- The proposal distribution is  $\mathcal{N}(x, \Gamma)$ .
- As in the scalar case, either too “small” or too “large” a  $\Gamma$  leads to poor results.

## 2.2 The Finnish example: the symmetric random walk Metropolis (SRWM) algorithm

- We consider the Metropolis algorithm, here in a multivariate context.
- The proposal distribution is  $\mathcal{N}(x, \Gamma)$ .
- As in the scalar case, either too “small” or too “large” a  $\Gamma$  leads to poor results.
- It is shown in [Gelman Roberts Gilks 1995] that a good  $\Gamma$  is  $\lambda\Gamma_\pi$ , where

## 2.2 The Finnish example: the symmetric random walk Metropolis (SRWM) algorithm

- We consider the Metropolis algorithm, here in a multivariate context.
- The proposal distribution is  $\mathcal{N}(x, \Gamma)$ .
- As in the scalar case, either too “small” or too “large” a  $\Gamma$  leads to poor results.
- It is shown in [Gelman Roberts Gilks 1995] that a good  $\Gamma$  is  $\lambda\Gamma_\pi$ , where
  - $\lambda = 2.38^2/n_x$ .

## 2.2 The Finnish example: the symmetric random walk Metropolis (SRWM) algorithm

- We consider the Metropolis algorithm, here in a multivariate context.
- The proposal distribution is  $\mathcal{N}(x, \Gamma)$ .
- As in the scalar case, either too “small” or too “large” a  $\Gamma$  leads to poor results.
- It is shown in [Gelman Roberts Gilks 1995] that a good  $\Gamma$  is  $\lambda\Gamma_\pi$ , where
  - $\lambda = 2.38^2/n_x$ .
  - $\Gamma_\pi$  is the covariance matrix of  $\pi$ , unknown a priori!

## 2.3 Example : “Learning” $\Gamma_\pi$

## 2.3 Example : “Learning” $\Gamma_\pi$

[Haario Saksman Tamminen 2001] have proposed to “learn  $\Gamma$  on the fly”.

## 2.3 Example : “Learning” $\Gamma_\pi$

[Haario Saksman Tamminen 2001] have proposed to “learn  $\Gamma$  on the fly”. It proceeds as follows:

## 2.3 Example : “Learning” $\Gamma_\pi$

[Haario Saksman Tamminen 2001] have proposed to “learn  $\Gamma$  on the fly”. It proceeds as follows:

At iteration  $k + 1$  of the Metropolis algorithm, given an estimate  $\mu_k, \Gamma_k$  constructed from  $X_1, \dots, X_k$ :

## 2.3 Example : “Learning” $\Gamma_\pi$

[Haario Saksman Tamminen 2001] have proposed to “learn  $\Gamma$  on the fly”. It proceeds as follows:

At iteration  $k + 1$  of the Metropolis algorithm, given an estimate  $\mu_k, \Gamma_k$  constructed from  $X_1, \dots, X_k$ :

1. Sample  $X_{k+1} \sim P_{\mathcal{N}(X_k, \lambda \Gamma_k)}^{SRWM}$ .

## 2.3 Example : “Learning” $\Gamma_\pi$

[Haario Saksman Tamminen 2001] have proposed to “learn  $\Gamma$  on the fly”. It proceeds as follows:

At iteration  $k + 1$  of the Metropolis algorithm, given an estimate  $\mu_k, \Gamma_k$  constructed from  $X_1, \dots, X_k$ :

1. Sample  $X_{k+1} \sim P_{\mathcal{N}(X_k, \lambda \Gamma_k)}^{SRWM}$ .
2. Set  $\gamma_{k+1} = 1/(k + 1)$  and update  $\mu_k, \Gamma_k$

## 2.3 Example : “Learning” $\Gamma_\pi$

[Haario Saksman Tamminen 2001] have proposed to “learn  $\Gamma$  on the fly”. It proceeds as follows:

At iteration  $k + 1$  of the Metropolis algorithm, given an estimate  $\mu_k, \Gamma_k$  constructed from  $X_1, \dots, X_k$ :

1. Sample  $X_{k+1} \sim P_{\mathcal{N}(X_k, \lambda \Gamma_k)}^{SRWM}$ .
2. Set  $\gamma_{k+1} = 1/(k + 1)$  and update  $\mu_k, \Gamma_k$

$$\mu_{k+1}$$

## 2.3 Example : “Learning” $\Gamma_\pi$

[Haario Saksman Tamminen 2001] have proposed to “learn  $\Gamma$  on the fly”. It proceeds as follows:

At iteration  $k + 1$  of the Metropolis algorithm, given an estimate  $\mu_k, \Gamma_k$  constructed from  $X_1, \dots, X_k$ :

1. Sample  $X_{k+1} \sim P_{\mathcal{N}(X_k, \lambda \Gamma_k)}^{SRWM}$ .
2. Set  $\gamma_{k+1} = 1/(k + 1)$  and update  $\mu_k, \Gamma_k$

$$\mu_{k+1} = (1 - \gamma_{k+1})\mu_k + \gamma_{k+1}X_{k+1}$$

## 2.3 Example : “Learning” $\Gamma_\pi$

[Haario Saksman Tamminen 2001] have proposed to “learn  $\Gamma$  on the fly”. It proceeds as follows:

At iteration  $k + 1$  of the Metropolis algorithm, given an estimate  $\mu_k, \Gamma_k$  constructed from  $X_1, \dots, X_k$ :

1. Sample  $X_{k+1} \sim P_{\mathcal{N}(X_k, \lambda \Gamma_k)}^{SRWM}$ .
2. Set  $\gamma_{k+1} = 1/(k + 1)$  and update  $\mu_k, \Gamma_k$

$$\begin{aligned}\mu_{k+1} &= (1 - \gamma_{k+1})\mu_k + \gamma_{k+1}X_{k+1} \\ &= \mu_k + \gamma_{k+1}(X_{k+1} - \mu_k)\end{aligned}$$

One can rewrite the update for  $(\mu_{k+1}, \Gamma_{k+1})$  as follows,

$$\mu_{k+1} = \mu_k + \gamma_{k+1}(X_{k+1} - \mu_k)$$

$$\Gamma_{k+1} = \Gamma_k + \gamma_{k+1}((X_{k+1} - \mu_k)(X_{k+1} - \mu_k)^\top - \Gamma_k)$$

One can rewrite the update for  $(\mu_{k+1}, \Gamma_{k+1})$  as follows,

$$\mu_{k+1} = \mu_k + \gamma_{k+1}(X_{k+1} - \mu_k)$$

$$\Gamma_{k+1} = \Gamma_k + \gamma_{k+1}((X_{k+1} - \mu_k)(X_{k+1} - \mu_k)^\top - \Gamma_k)$$

i.e. with  $\theta_{k+1} := (\mu_{k+1}, \Gamma_{k+1})$

$$\theta_{k+1} = \theta_k + \gamma_{k+1}H(\theta_k, X_{k+1})$$

One can rewrite the update for  $(\mu_{k+1}, \Gamma_{k+1})$  as follows,

$$\mu_{k+1} = \mu_k + \gamma_{k+1}(X_{k+1} - \mu_k)$$

$$\Gamma_{k+1} = \Gamma_k + \gamma_{k+1}((X_{k+1} - \mu_k)(X_{k+1} - \mu_k)^\top - \Gamma_k)$$

i.e. with  $\theta_{k+1} := (\mu_{k+1}, \Gamma_{k+1})$

$$\theta_{k+1} = \theta_k + \gamma_{k+1}H(\theta_k, X_{k+1})$$

where for  $\theta = (\mu, \Gamma)$

$$H(\theta, X) := \left( X - \mu, (X - \mu)(X - \mu)^\top - \Gamma \right).$$

One can rewrite the update for  $(\mu_{k+1}, \Gamma_{k+1})$  as follows,

$$\mu_{k+1} = \mu_k + \gamma_{k+1}(X_{k+1} - \mu_k)$$

$$\Gamma_{k+1} = \Gamma_k + \gamma_{k+1}((X_{k+1} - \mu_k)(X_{k+1} - \mu_k)^\top - \Gamma_k)$$

i.e. with  $\theta_{k+1} := (\mu_{k+1}, \Gamma_{k+1})$

$$\theta_{k+1} = \theta_k + \gamma_{k+1}H(\theta_k, X_{k+1})$$

where for  $\theta = (\mu, \Gamma)$

$$H(\theta, X) := \left( X - \mu, (X - \mu)(X - \mu)^\top - \Gamma \right).$$

As pointed out in [Andrieu & Robert 2001] this is a noisy gradient algorithm.

## 2.4 Coerced acceptance ratio of MH

- Consider again the scalar SRWM algorithm with  $\mathcal{N}(x, \theta^2)$  as proposal.

## 2.4 Coerced acceptance ratio of MH

- Consider again the scalar SRWM algorithm with  $\mathcal{N}(x, \theta^2)$  as proposal.
- The expected acceptance probability in the stationary regime is

## 2.4 Coerced acceptance ratio of MH

- Consider again the scalar SRWM algorithm with  $\mathcal{N}(x, \theta^2)$  as proposal.
- The expected acceptance probability in the stationary regime is

$$\tau(\theta)$$

## 2.4 Coerced acceptance ratio of MH

- Consider again the scalar SRWM algorithm with  $\mathcal{N}(x, \theta^2)$  as proposal.
- The expected acceptance probability in the stationary regime is

$$\tau(\theta) = \iint_{\mathbf{X} \times \mathbf{X}} \alpha(x, y)$$

## 2.4 Coerced acceptance ratio of MH

- Consider again the scalar SRWM algorithm with  $\mathcal{N}(x, \theta^2)$  as proposal.
- The expected acceptance probability in the stationary regime is

$$\tau(\theta) = \iint_{\mathcal{X} \times \mathcal{X}} \alpha(x, y) \pi(x)$$

## 2.4 Coerced acceptance ratio of MH

- Consider again the scalar SRWM algorithm with  $\mathcal{N}(x, \theta^2)$  as proposal.
- The expected acceptance probability in the stationary regime is

$$\tau(\theta) = \iint_{\mathbf{X} \times \mathbf{X}} \alpha(x, y) \pi(x) q_{\theta}(x, y)$$

## 2.4 Coerced acceptance ratio of MH

- Consider again the scalar SRWM algorithm with  $\mathcal{N}(x, \theta^2)$  as proposal.
- The expected acceptance probability in the stationary regime is

$$\tau(\theta) = \iint_{\mathbf{X} \times \mathbf{X}} \alpha(x, y) \pi(x) q_{\theta}(x, y) dx dy$$

## 2.4 Coerced acceptance ratio of MH

- Consider again the scalar SRWM algorithm with  $\mathcal{N}(x, \theta^2)$  as proposal.
- The expected acceptance probability in the stationary regime is

$$\begin{aligned}\tau(\theta) &= \iint_{\mathbf{X} \times \mathbf{X}} \alpha(x, y) \pi(x) q_{\theta}(x, y) dx dy \\ &= E_{\pi \times q_{\theta}}(\alpha(X, Y))\end{aligned}$$

## 2.4 Coerced acceptance ratio of MH

- Consider again the scalar SRWM algorithm with  $\mathcal{N}(x, \theta^2)$  as proposal.
- The expected acceptance probability in the stationary regime is

$$\begin{aligned}\tau(\theta) &= \iint_{\mathbf{X} \times \mathbf{X}} \alpha(x, y) \pi(x) q_\theta(x, y) dx dy \\ &= E_{\pi \times q_\theta}(\alpha(X, Y))\end{aligned}$$

- In [Roberts & Rosenthal, 2001], it is shown that if  $n_x \rightarrow \infty$  then for some  $\pi$ 's the optimal  $\theta_*$  is such that  $\tau(\theta_*) = 0.234$ .

- If  $\pi$  is such that  $\tau(\theta)$  is a decreasing function of  $\theta$ .

- If  $\pi$  is such that  $\tau(\theta)$  is a decreasing function of  $\theta$ .
- Then

$$h(\theta)$$

- If  $\pi$  is such that  $\tau(\theta)$  is a decreasing function of  $\theta$ .
- Then

$$h(\theta) = \tau(\theta) - \tau^*$$

- If  $\pi$  is such that  $\tau(\theta)$  is a decreasing function of  $\theta$ .
- Then

$$\begin{aligned}h(\theta) &= \tau(\theta) - \tau^* \\ &= E_{\pi \times q_\theta} (\alpha(X, Y) - \tau^*)\end{aligned}$$

- If  $\pi$  is such that  $\tau(\theta)$  is a decreasing function of  $\theta$ .
- Then

$$\begin{aligned}h(\theta) &= \tau(\theta) - \tau^* \\ &= E_{\pi \times q_\theta} (\alpha(X, Y) - \tau^*)\end{aligned}$$

is a decreasing function of  $\theta$ .

- If  $\pi$  is such that  $\tau(\theta)$  is a decreasing function of  $\theta$ .
- Then

$$\begin{aligned}h(\theta) &= \tau(\theta) - \tau^* \\ &= E_{\pi \times q_\theta} (\alpha(X, Y) - \tau^*)\end{aligned}$$

is a decreasing function of  $\theta$ .

- And one can suggest the following noisy gradient algorithm.

At iteration  $i + 1$

$$Y_{k+1} \sim q_{\theta_k}(X_k, \cdot)$$

$$X_{k+1} = \begin{cases} Y_{k+1} & \text{with probability } \alpha(X_k, Y_{k+1}) \\ X_k & \text{otherwise} \end{cases}$$

$$\theta_{k+1} = \theta_k + \gamma_{k+1}(\alpha(X_k, Y_{k+1}) - \tau^*)$$

At iteration  $i + 1$

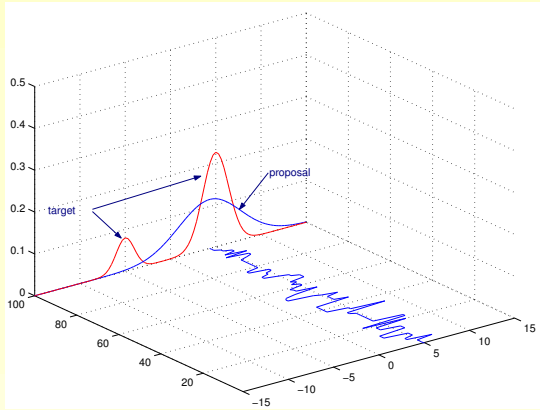
$$Y_{k+1} \sim q_{\theta_k}(X_k, \cdot)$$

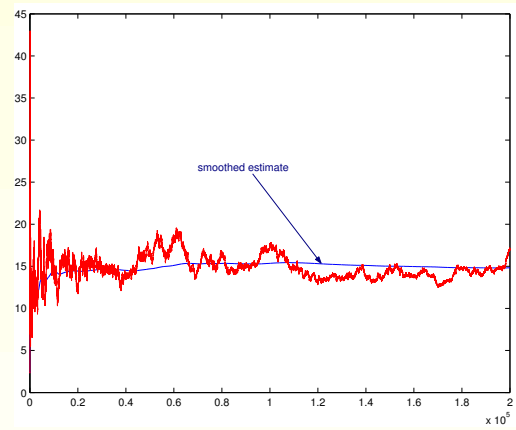
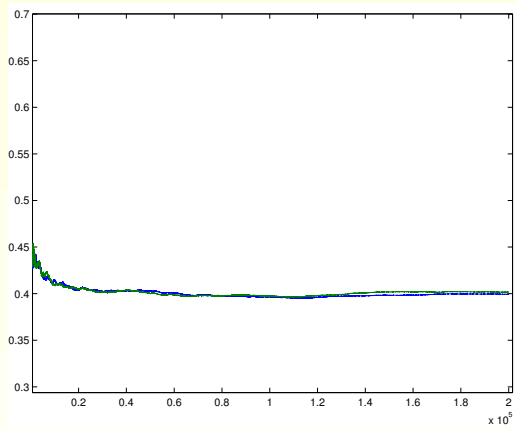
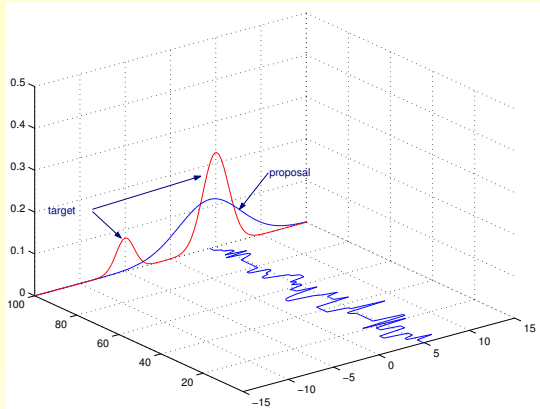
$$X_{k+1} = \begin{cases} Y_{k+1} & \text{with probability } \alpha(X_k, Y_{k+1}) \\ X_k & \text{otherwise} \end{cases}$$

$$\theta_{k+1} = \theta_k + \gamma_{k+1}(\alpha(X_k, Y_{k+1}) - \tau^*)$$

that is here the aim is to find the zeroes of

$$\begin{aligned} h(\theta) &= \tau(\theta) - \tau^* \\ &= E_{\pi \times q_{\theta}}(\alpha(X, Y) - \tau^*). \end{aligned}$$





## 2.5 Fitting the target distribution

- The independent MH uses  $q_{\theta}(y)$ , i.e. independent of the past.

## 2.5 Fitting the target distribution

- The independent MH uses  $q_{\theta}(y)$ , i.e. independent of the past.
- It is known to work well when  $q_{\theta}$  is “similar” to  $\pi$ .

## 2.5 Fitting the target distribution

- The independent MH uses  $q_{\theta}(y)$ , i.e. independent of the past.
- It is known to work well when  $q_{\theta}$  is “similar” to  $\pi$ .
- One can suggest minimizing some “distance” between  $\pi$  and  $q_{\theta}$  (e.g. the Kullback-Leibler divergence).

## 2.6 Fitting the banana distribution

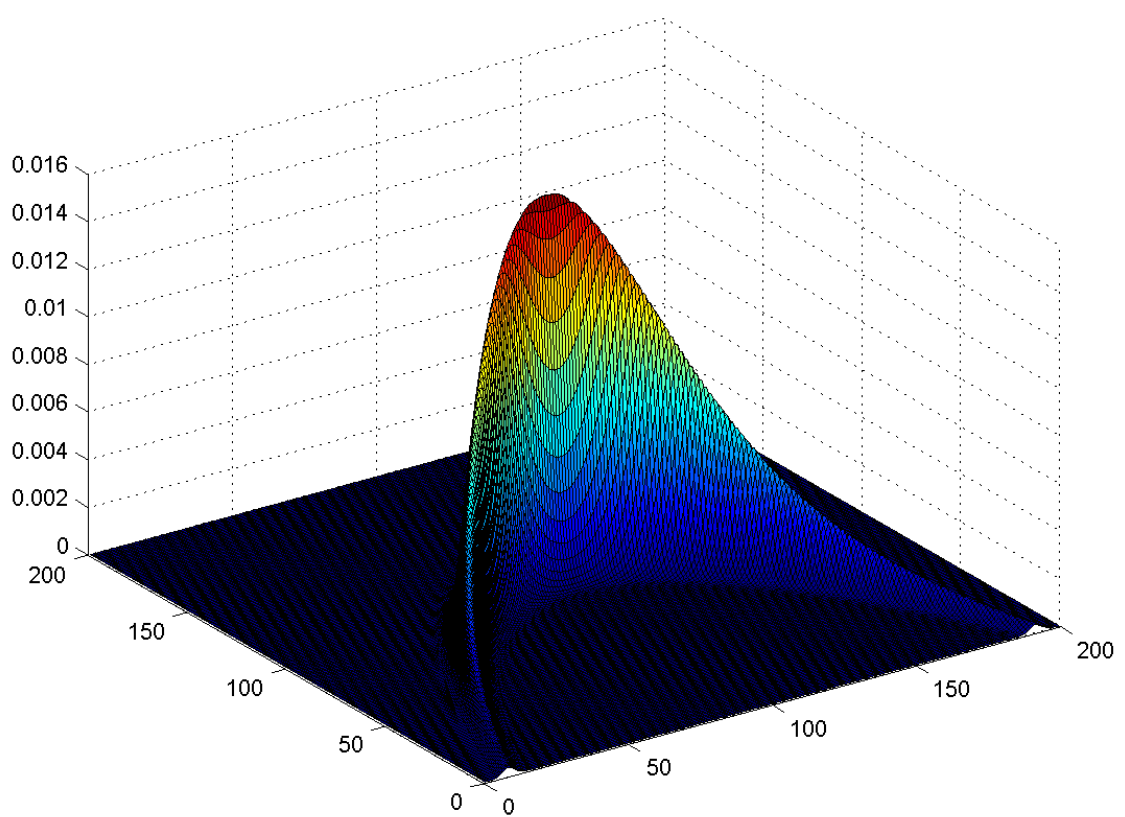
- We consider a target distribution with non linear correlation.

## 2.6 Fitting the banana distribution

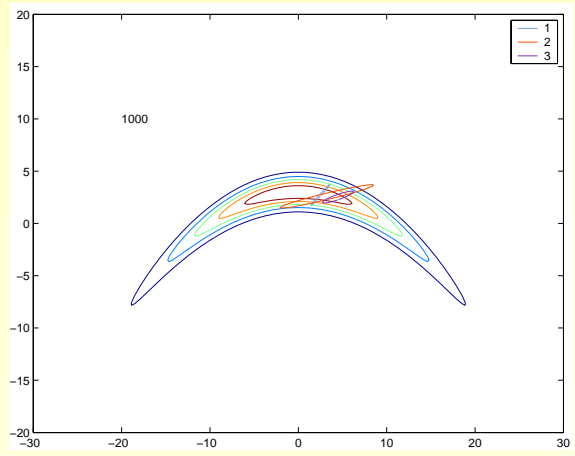
- We consider a target distribution with non linear correlation.
- We aim at fitting it with a mixture of three normal distributions.

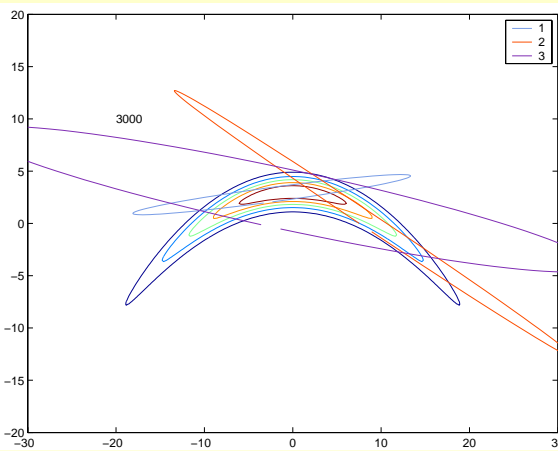
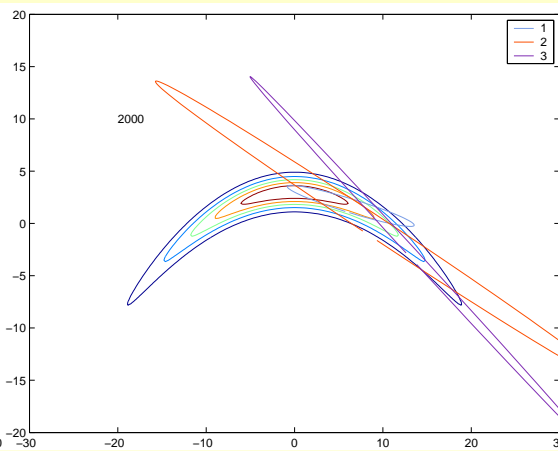
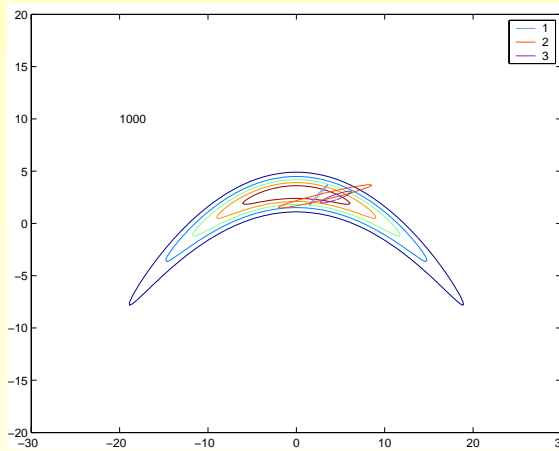
## 2.6 Fitting the banana distribution

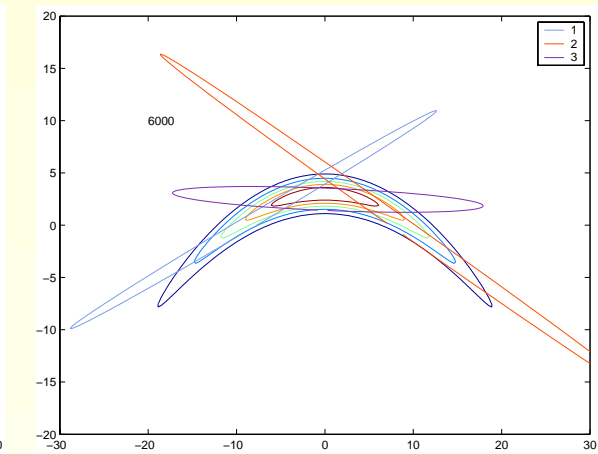
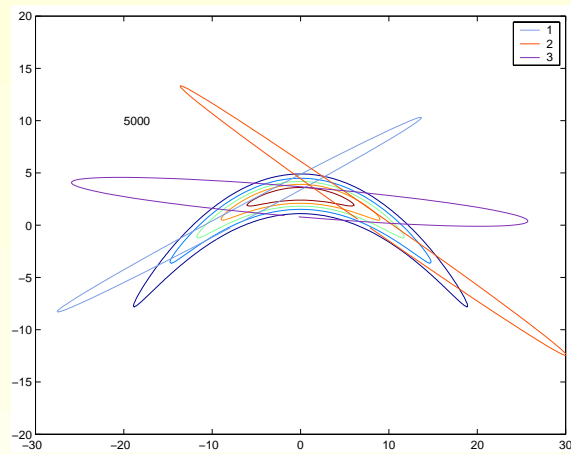
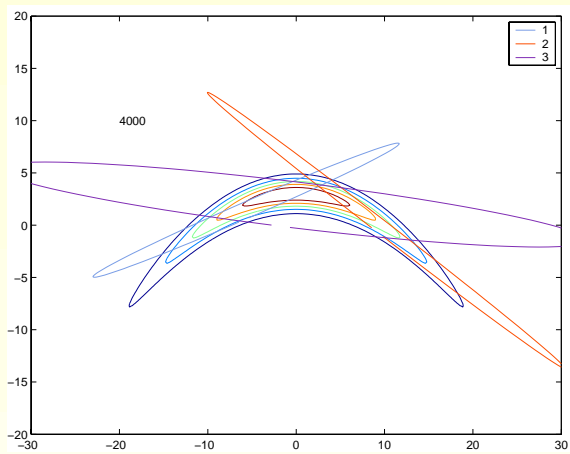
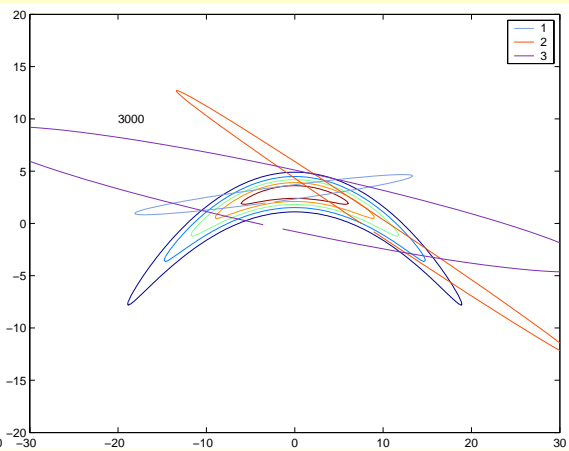
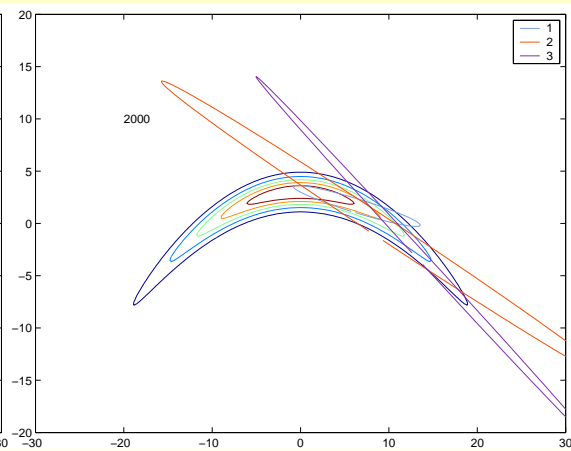
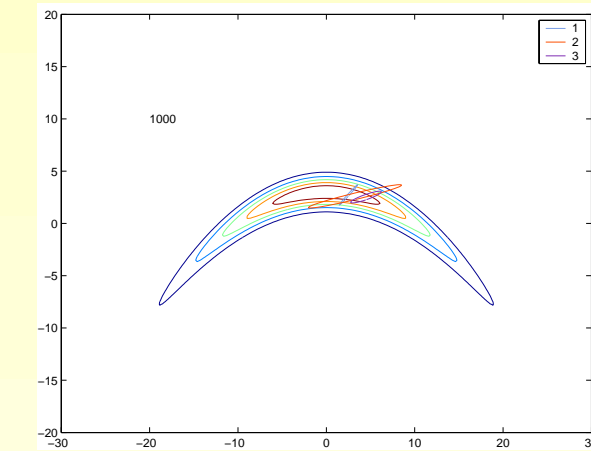
- We consider a target distribution with non linear correlation.
- We aim at fitting it with a mixture of three normal distributions.
- The parameters are unknown, and fitted using an on-line version of the EM algorithm.

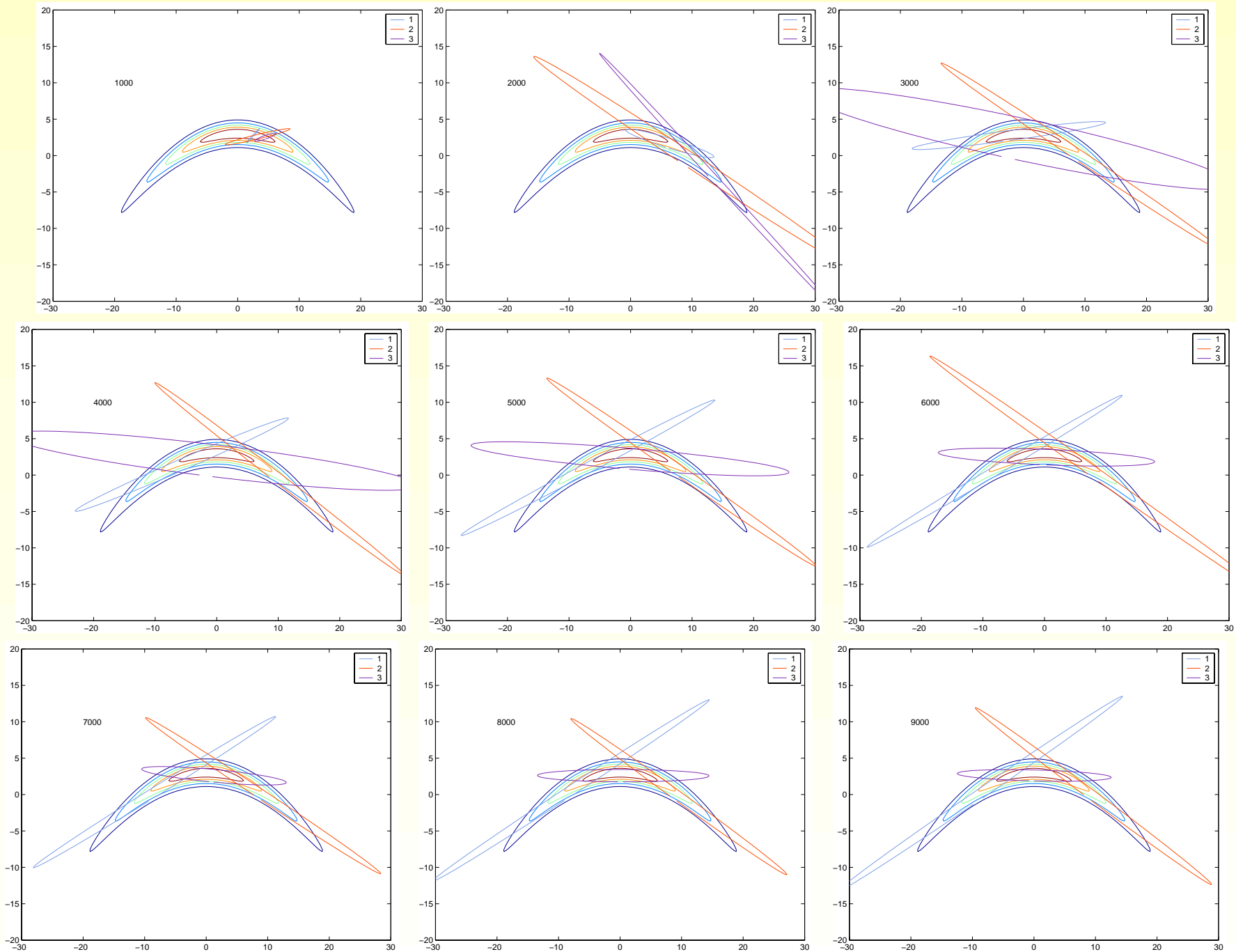












## 3 Theoretical perspective

### 3.1 What can go wrong: a toy example.

- Due to the coupling of  $\{\theta_i\}$  and  $\{X_i\}$ , the **ergodicity** properties of  $\{X_i\}$  do not straightforwardly follow from standard Markov chain argument

## 3 Theoretical perspective

### 3.1 What can go wrong: a toy example.

- Due to the coupling of  $\{\theta_i\}$  and  $\{X_i\}$ , the **ergodicity** properties of  $\{X_i\}$  do not straightforwardly follow from standard Markov chain argument
- Consider the following toy example with  $X = \{1, 2\}$  and transition probability

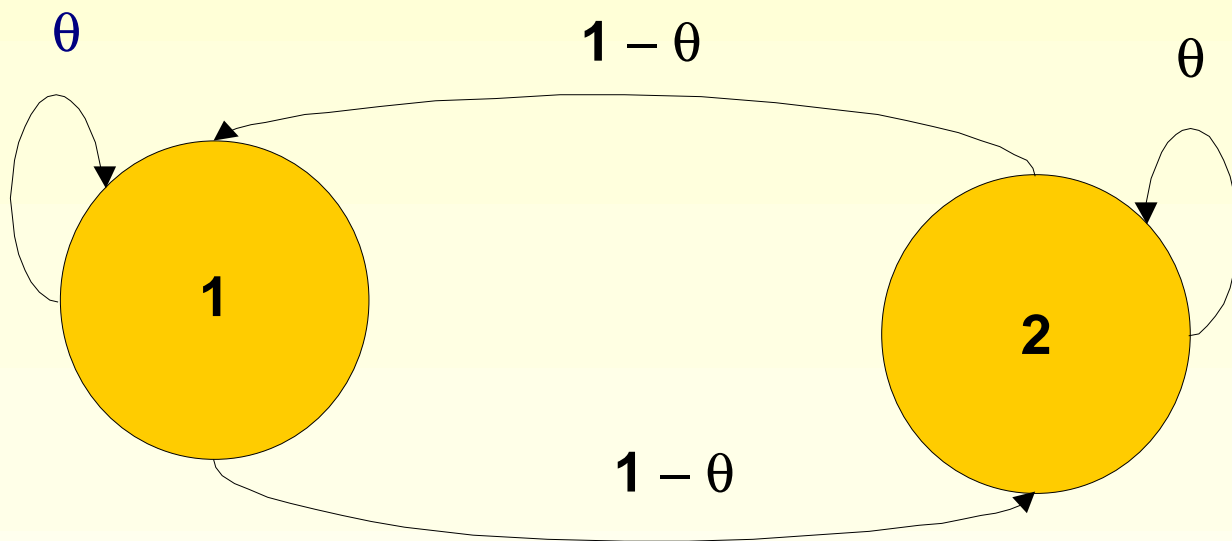
$$P_\theta = \begin{bmatrix} \theta & 1 - \theta \\ 1 - \theta & \theta \end{bmatrix},$$

## 3 Theoretical perspective

### 3.1 What can go wrong: a toy example.

- Due to the coupling of  $\{\theta_i\}$  and  $\{X_i\}$ , the **ergodicity** properties of  $\{X_i\}$  do not straightforwardly follow from standard Markov chain argument
- Consider the following toy example with  $X = \{1, 2\}$  and transition probability

$$P_\theta = \begin{bmatrix} \theta & 1 - \theta \\ 1 - \theta & \theta \end{bmatrix},$$



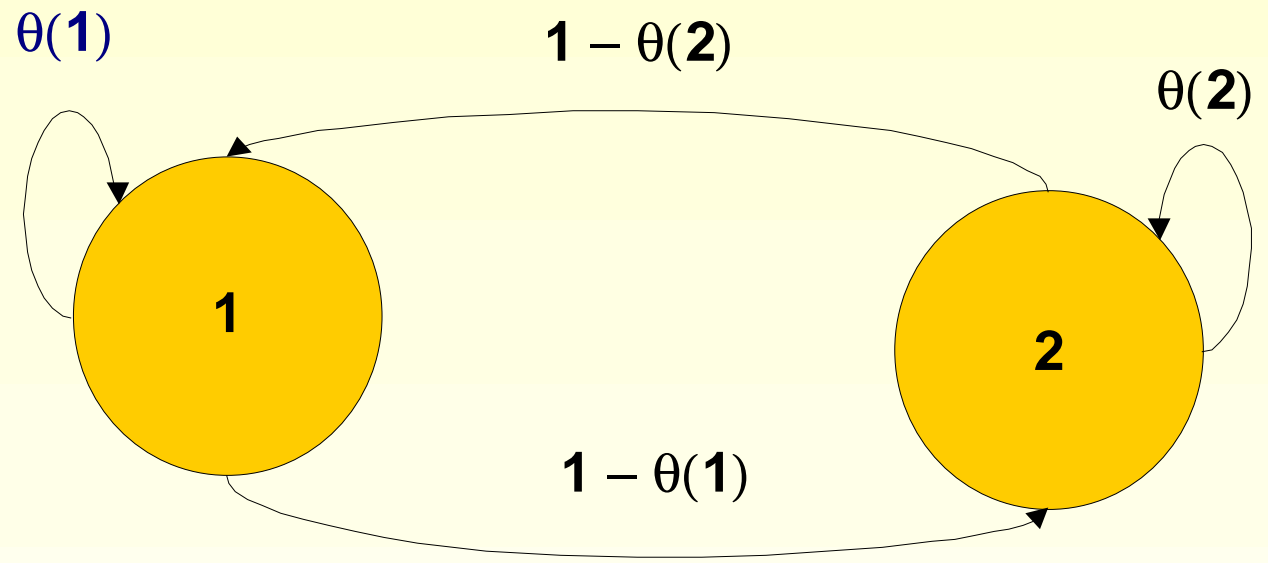
- Not surprisingly for any  $\theta \in [0, 1]$ ,  $P_\theta$  has  $\pi = ( 1/2 \ 1/2 )$  as invariant distribution.

- Now assume that at iteration  $i + 1$ ,  $\theta$  is a function  $\theta(X_i)$ , the previous state of the Markov chain,

- Now assume that at iteration  $i + 1$ ,  $\theta$  is a function  $\theta(X_i)$ , the previous state of the Markov chain,
- The process is **still** a Markov chain,

- Now assume that at iteration  $i + 1$ ,  $\theta$  is a function  $\theta(X_i)$ , the previous state of the Markov chain,
- The process is **still** a Markov chain,
- But, denoting  $\theta_1 = \theta(1)$  and  $\theta_2 = \theta(2)$ ,

- Now assume that at iteration  $i + 1$ ,  $\theta$  is a function  $\theta(X_i)$ , the previous state of the Markov chain,
- The process is **still** a Markov chain,
- But, denoting  $\theta_1 = \theta(1)$  and  $\theta_2 = \theta(2)$ ,



- The invariant distribution of  $\tilde{P}$  can be shown to be

- The invariant distribution of  $\tilde{P}$  can be shown to be

$$\mu_1 = \frac{\theta_1}{\theta_1 + \theta_2} \quad \mu_2 = \frac{\theta_2}{\theta_1 + \theta_2}$$

- The invariant distribution of  $\tilde{P}$  can be shown to be

$$\mu_1 = \frac{\theta_1}{\theta_1 + \theta_2} \quad \mu_2 = \frac{\theta_2}{\theta_1 + \theta_2}$$

Invariance of  $\pi = (1/2, 1/2)$  is lost.

- The invariant distribution of  $\tilde{P}$  can be shown to be

$$\mu_1 = \frac{\theta_1}{\theta_1 + \theta_2} \quad \mu_2 = \frac{\theta_2}{\theta_1 + \theta_2}$$

Invariance of  $\pi = (1/2, 1/2)$  is lost.

- BUT, if the dependence on the current state vanishes with time, then we might eventually recover  $\pi = (1/2, 1/2)$ .

## 3.2 A result.

- There exist constants  $A(\epsilon, \mathcal{K})$  and  $B(\epsilon, \mathcal{K})$  such that for any  $n \geq 1$ ,

## 3.2 A result.

- There exist constants  $A(\epsilon, \mathcal{K})$  and  $B(\epsilon, \mathcal{K})$  such that for any  $n \geq 1$ ,

$$\sqrt{E \left| \hat{E}_N \right|^2} \leq \frac{A(\epsilon, \mathcal{K})}{\sqrt{N}} + B(\epsilon, \mathcal{K}) \frac{\sum_{k=1}^N \epsilon_k}{N}.$$

## 3.2 A result.

- There exist constants  $A(\epsilon, \mathcal{K})$  and  $B(\epsilon, \mathcal{K})$  such that for any  $n \geq 1$ ,

$$\sqrt{E \left| \hat{E}_N \right|^2} \leq \frac{A(\epsilon, \mathcal{K})}{\sqrt{N}} + B(\epsilon, \mathcal{K}) \frac{\sum_{k=1}^N \epsilon_k}{N}.$$

1. The first term corresponds to the Monte Carlo fluctuations.

## 3.2 A result.

- There exist constants  $A(\epsilon, \mathcal{K})$  and  $B(\epsilon, \mathcal{K})$  such that for any  $n \geq 1$ ,

$$\sqrt{E \left| \hat{E}_N \right|^2} \leq \frac{A(\epsilon, \mathcal{K})}{\sqrt{N}} + B(\epsilon, \mathcal{K}) \frac{\sum_{k=1}^N \epsilon_k}{N}.$$

1. The first term corresponds to the Monte Carlo fluctuations.
2. The **second term** is the price to pay for adaptation.

## 3.2 A result.

- There exist constants  $A(\epsilon, \mathcal{K})$  and  $B(\epsilon, \mathcal{K})$  such that for any  $n \geq 1$ ,

$$\sqrt{E \left| \hat{E}_N \right|^2} \leq \frac{A(\epsilon, \mathcal{K})}{\sqrt{N}} + B(\epsilon, \mathcal{K}) \frac{\sum_{k=1}^N \epsilon_k}{N}.$$

1. The first term corresponds to the Monte Carlo fluctuations.
  2. The **second term** is the price to pay for adaptation.
- More can be said about the second term.

$$\sqrt{E |E_N|^2} \leq \frac{A(\epsilon, \mathcal{K})}{\sqrt{N}} + B(\epsilon, \mathcal{K}) \frac{\sum_{k=1}^N \epsilon_k}{N}.$$

$$\sqrt{E |E_N|^2} \leq \frac{A(\epsilon, \mathcal{K})}{\sqrt{N}} + B(\epsilon, \mathcal{K}) \frac{\sum_{k=1}^N \epsilon_k}{N}.$$

Assume that  $\epsilon_i = i^{-\gamma}$  for  $\gamma \in (0, 1)$ , then

$$\sqrt{E |E_N|^2} \leq \frac{A(\epsilon, \mathcal{K})}{\sqrt{N}} + B(\epsilon, \mathcal{K}) \frac{\sum_{k=1}^N \epsilon_k}{N}.$$

Assume that  $\epsilon_i = i^{-\gamma}$  for  $\gamma \in (0, 1)$ , then

$$\frac{\sum_{k=1}^N \epsilon_k}{N} \sim \frac{1}{1-\gamma} N^{-\gamma},$$

$$\sqrt{E |E_N|^2} \leq \frac{A(\epsilon, \mathcal{K})}{\sqrt{N}} + B(\epsilon, \mathcal{K}) \frac{\sum_{k=1}^N \epsilon_k}{N}.$$

Assume that  $\epsilon_i = i^{-\gamma}$  for  $\gamma \in (0, 1)$ , then

$$\frac{\sum_{k=1}^N \epsilon_k}{N} \sim \frac{1}{1-\gamma} N^{-\gamma},$$

and for  $\gamma \in (1/2, 1]$  the second term asymptotically vanishes.

## 4 Conclusions

- Opportunities for further computer aided design of efficient MCMC algorithms within existing packages.

## 4 Conclusions

- Opportunities for further computer aided design of efficient MCMC algorithms within existing packages.
- Some theory justifies, to some extent, the use of vanishing adaptation.

## 4 Conclusions

- Opportunities for further computer aided design of efficient MCMC algorithms within existing packages.
- Some theory justifies, to some extent, the use of vanishing adaptation.
- Usually the modification of standard MCMC code is minor,

## 4 Conclusions

- Opportunities for further computer aided design of efficient MCMC algorithms within existing packages.
- Some theory justifies, to some extent, the use of vanishing adaptation.
- Usually the modification of standard MCMC code is minor,
- Further reading [Erland 2003], [Atchadé & Rosenthal 2003], [Pasarica & Gelman 2003], [Doucet & Tadić 2002] [Hastie 2005], [Roberts & Rosenthal 2005].

## 4 Conclusions

- Opportunities for further computer aided design of efficient MCMC algorithms within existing packages.
- Some theory justifies, to some extent, the use of vanishing adaptation.
- Usually the modification of standard MCMC code is minor,
- Further reading [Erland 2003], [Atchadé & Rosenthal 2003], [Pasarica & Gelman 2003], [Doucet & Tadić 2002] [Hastie 2005], [Roberts & Rosenthal 2005].

Thanks for your attention!