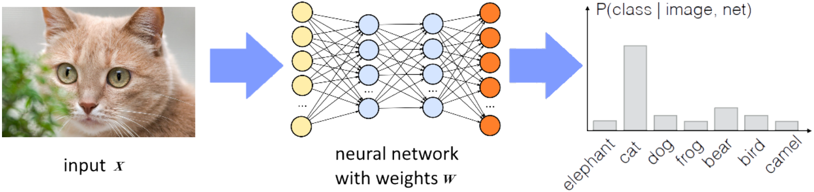
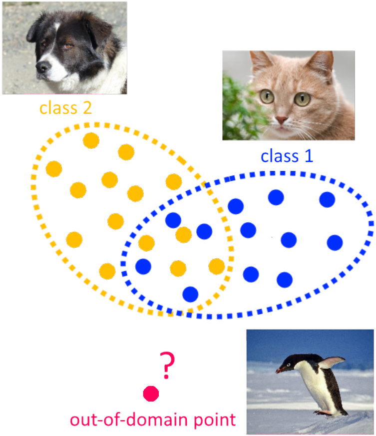
Convolutional networks



- Example: classification problem
- Output: $p(y|x, w)$ for each class y.

Uncertainty estimation

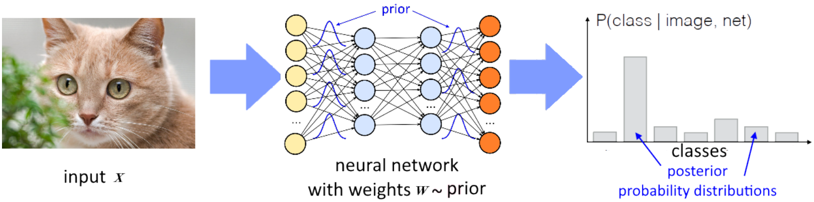


⇝ If input is far away from training data, instead of saying "I don't know", the network can be overconfident due to overfitting.

- We want to know the uncertainty of the predictions.

Bayesian deep learning[1]

- To account for uncertainty, the output of the NN is a *probability distribution* of … well, probability distributions.
- Random initialisations for $w \sim p$ some prior.



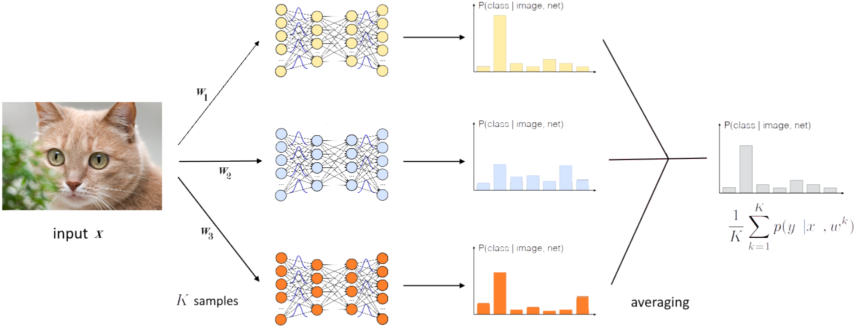
- Instead of point prediction $p(y|x,w)$, Bayesian $p(y|x,w)$ is a random variable, w-measurable.

$\rightsquigarrow$ Take the expectancy $\mathbb{E}_w[p(y|x,w)]$ for a prediction.

[1] inspired from *Stochastic variational inference and Bayesian neural networks* by Nadezhda Chirkova, HSE Moscow, Russia

BNN as a family of NN

- To estimate $\mathbb{E}_w[p(y|x, w)]$, take the average over several NN with different initial weights, sampled from the prior $p(w|X, Y) \rightsquigarrow$ more calculations.



Why go Bayesian?

This method allows to:

1. Limit overfitting, as an average of - opposite - overconfident guesses is closer to uniform = no guess.
2. Estimate uncertainty and reduce it by taking more averages.
3. Prior can come with properties we want to infer into the network. E.g. after one training, we can use the posterior distribution as a new prior to train again on another data set $\rightsquigarrow$ continual learning.

Bayesian considerations for training BNN

- The framework "prior x data $\rightsquigarrow$ posterior" is the classical Bayesian one.
- We need to access the posterior (renormalized Likelihood x Prior)

$$p(w|X, Y) = \frac{p(Y|X, w)p(w)}{\int p(Y|X, \tilde{w})p(\tilde{w})\mathrm{d}\tilde{w}} .$$

- In practice, the integral is intractable:
  - $\rightsquigarrow$ Usually we approximate the posterior with a well known parametric family with stochastic variational inference[2].
  - $\rightsquigarrow$ We can also approximate this integral with MCMC methods.

[2] see *Bayesian neural networks* from Dmitry Molchanov, Samsung AI, Moscow Russia

Bayesian considerations for training BNN

- The framework "prior x data $\rightsquigarrow$ posterior" is the classical Bayesian one.
- We need to access the posterior (renormalized Likelihood x Prior)

$$p(w|X,Y) = \frac{p(Y|X,w)p(w)}{\int p(Y|X,\tilde{w})p(\tilde{w})\mathrm{d}\tilde{w}} .$$

- In practice, the integral is intractable:
  - $\rightsquigarrow$ Usually we approximate the posterior with a well known parametric family with stochastic variational inference[2].
  - $\rightsquigarrow$ We can also approximate this integral with MCMC methods.
    - ✓ This is where my study can be useful.

[2] see *Bayesian neural networks* from Dmitry Molchanov, Samsung AI, Moscow Russia

Optimal Scaling of Metropolis adjusted Langevin approximations with Laplace target

Pablo Jiménez
under the supervision of Alain Durmus

December 11 2019

ENS Paris Saclay, CMLA

Centre de Mathématiques et de Leurs Applications

Outline

Motivation

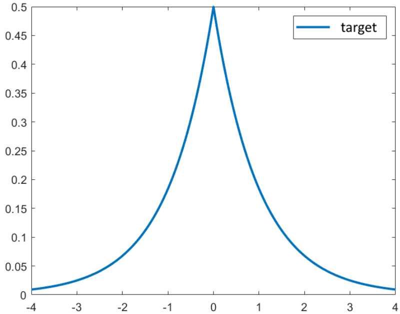
Metropolis-Hastings algorithm

The optimal scaling problem

Main results

Motivation

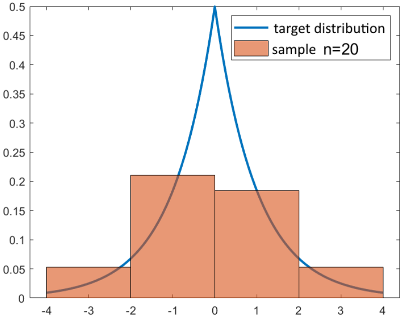
Sampling from a probability distribution



- Input: target density π over $\mathbb{R}^d$.
- Output: $(X_1^d, \ldots, X_n^d)_n$ i.i.d. sampling π.
- Goal: approximate integrals of the form $\int_{\mathbb{R}^d} h(x^d)\pi(x^d)\mathrm{d}x^d$.

See an example

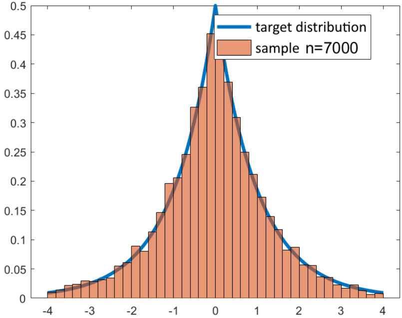
Sampling from a probability distribution



- Input: target density π over $\mathbb{R}^d$.
- Output: $(X_1^d, \ldots, X_n^d)_n$ i.i.d. sampling π.
- Goal: approximate integrals of the form $\int_{\mathbb{R}^d} h(x^d)\pi(x^d)\mathrm{d}x^d$.

See an example

Sampling from a probability distribution



- Input: target density π over $\mathbb{R}^d$.
- Output: $(X_1^d, \ldots, X_n^d)_n$ i.i.d. sampling π.
- Goal: approximate integrals of the form $\int_{\mathbb{R}^d} h(x^d)\pi(x^d)\mathrm{d}x^d$.

See an example

Monte Carlo method: summary

- Goal: approximate integrals of the form $\int_{\mathbb{R}^d} h(x^d)\pi(x^d)\mathrm{d}x^d$.
- Estimate with $n^{-1}\sum_{k=1}^{n} h(X_k^d)$.
- Useful in high dimension $\longrightarrow$ Bayesian inference.
- Problem: $(X_n^d)_{\mathrm{n}\in\mathbb{N}^*}$ i.i.d. of law π $\longrightarrow$ costly/intractable.

Monte Carlo method: summary

- Goal: approximate integrals of the form $\int_{\mathbb{R}^d} h(x^d)\pi(x^d)\mathrm{d}x^d$.
- Estimate with $n^{-1}\sum_{k=1}^{n} h(X_k^d)$.
- Useful in high dimension $\longrightarrow$ Bayesian inference.
- Problem: $(X_n^d)_{\mathrm{n}\in\mathbb{N}^*}$ i.i.d. of law π $\longrightarrow$ costly/intractable.
- Solution: $(X_n^d)_{n\in\mathbb{N}^*}$ Markov chain approximating π.

Markov chain Monte Carlo (MCMC)

N. Metropolis
(1915-1999)

- Metropolis et al. (1953)
- Hastings (1970)

W.K. Hastings
(1930-2016)

Metropolis-Hastings algorithm

Metropolis-Hastings description

1. Given $X_n^d \in \mathbb{R}^d$, sample from the proposal Y_{n+1}^d under $q^d(X_n^d, \cdot)$.
2. Calculate the acceptance rate $\alpha(X_n^d, Y_{n+1}^d)$, where

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

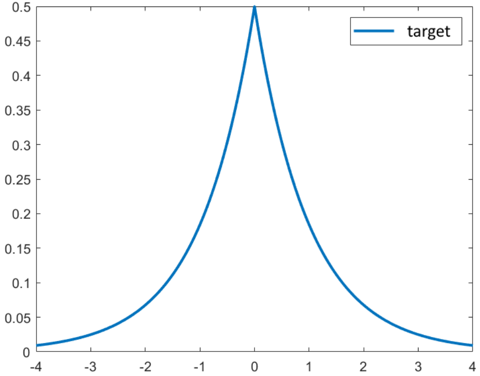
3. Sample U_{n+1}, uniform over $[0, 1]$.
4. Evaluate the acceptance event A_{n+1}^d, given by

$$\mathrm{A}_{n+1}^d = \left\{ U_{n+1} \leqslant \alpha(X_n^d, Y_{n+1}^d) \right\} .$$

5. Summary: X_{n+1}^d is given by

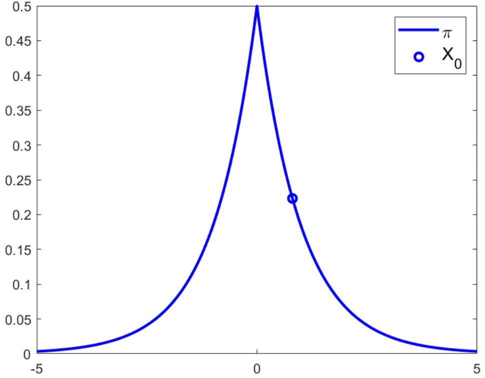
$$X_{n+1}^d = \underbrace{Y_{n+1}^d \mathbb{1}_{\mathrm{A}_{n+1}^d}}_{\text{accept}} + \underbrace{X_n^d \left(1 - \mathbb{1}_{\mathrm{A}_{n+1}^d}\right)}_{\text{reject}} .$$

Mise en pratique de l'algorithme



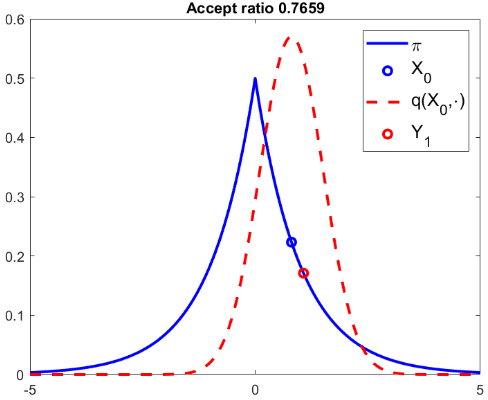
1. Sample X_0^d.
2. Sample Y_1^d.
3. Compute the acceptance probability $\alpha(X_n^d, Y_{n+1}^d)$.
4. Accept or reject Y_1^d .

Mise en pratique de l'algorithme



1. Sample X_0^d.
2. Sample Y_1^d.
3. Compute the acceptance probability $\alpha(X_n^d, Y_{n+1}^d)$.
4. Accept or reject Y_1^d .

Mise en pratique de l'algorithme

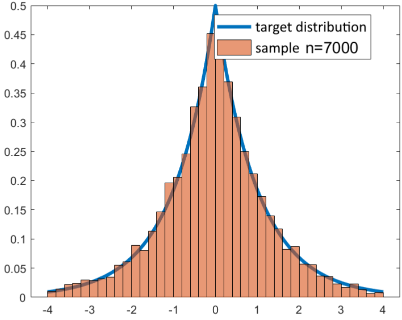


1. Sample X_0^d.
2. Sample Y_1^d.
3. Compute the acceptance probability $\alpha(X_n^d, Y_{n+1}^d)$.
4. Accept or reject Y_1^d .

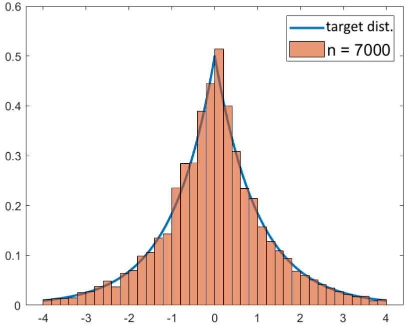
Metropolis-Hastings algorithm: results

Theorem : Tierney (1994)

Under reasonable assumptions on q^d, $(X_n^d)_n$ is an ergodic Markov chain, with invariant law π.



I.i.d. sample



Sample from a Markov chain

Metropolis-Hastings algorithm: results

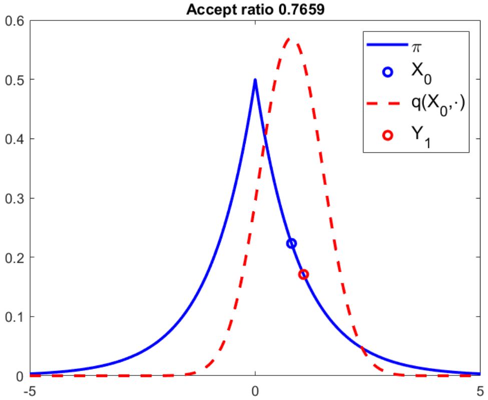
Theorem : Tierney (1994)

Under reasonable assumptions on q^d, $(X_n^d)_n$ is an ergodic Markov chain, with invariant law π.

Assumptions: q^d and π positives over $\mathbb{R}^d$.

How to choose the proposal q^d?

Example : random walk Metropolis (RWM)



Step of the random walk Metropolis algorithm.

- $Y^d_{n+1} = X^d_n + Z^d_{n+1}$, where Z^d_{n+1} is a zero-mean gaussian.
- Proposal $q^d(x^d, \cdot)$ zero-mean gaussian $\mathrm{N}(x^d, \sigma_d^2 \mathrm{Id}_d)$.
- Tune σ_d^2 w.r.t. the target π :

$\rightarrow$ large enough to limit correlations,

$\rightarrow$ small enough to have a high acceptance rate.

Example : Langevin dynamics

- Assume $\pi : x^d \mapsto \exp\left(-V(x^d)\right)/Z$, with normalization constant Z, and smooth $V : \mathbb{R}^d \to \mathbb{R}$.
- SDE of the diffusion process $(\mathbf{X}_t^d)_{t \geqslant 0}$:

$$\mathrm{d}\mathbf{X}_t^d = -\nabla V\left(\mathbf{X}_t^d\right)\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t^d\,,$$

  where $\mathbf{B}^d$ is a standard brownian motion over $\mathbb{R}^d$.
- Roberts and Tweedie (1996) : $(\mathbf{X}_t^d)_{t \geqslant 0}$ is ergodic with invariant distribution π.
- Problem: intractable in practice.
- Idea: use Euler-Murayama discretization.

Example : unadjusted Langevin approximation

- SDE of the diffusion process $(\mathbf{X}_t^d)_{t \geqslant 0}$:

$$\mathrm{d}\mathbf{X}_t^d = -\nabla V\left(\mathbf{X}_t^d\right)\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t^d \; .$$

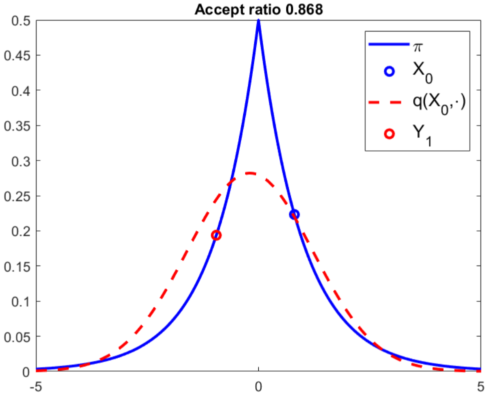
- Euler–Maruyama discretization $(\mathrm{X}_k^d)_{k \geqslant 0}$, with step-size $\sigma_d^2 > 0$ defines a Markov chain:

$$\mathrm{X}_{k+1}^d = \mathrm{X}_k^d - \sigma_d^2 \nabla V\left(\mathrm{X}_k^d\right) + \sqrt{2\sigma_d^2}\, Z_{k+1}^d \; ,$$

with independent sequences $\{(Z_k^d)_{k \geqslant 1} \mid d \in \mathbb{N}^*\}$, s.t. $(Z_k^d)_{k \geqslant 1}$ i.i.d. sampled from $\mathrm{N}(0, \mathrm{Id}_d)$.

- Problem : π is not invariant for $(\mathrm{X}_k^d)_{k \geqslant 1}$.
- Idea : introduce this dynamic as a proposal in M-H.

Metropolis adjusted Langevin approximations (MALA)



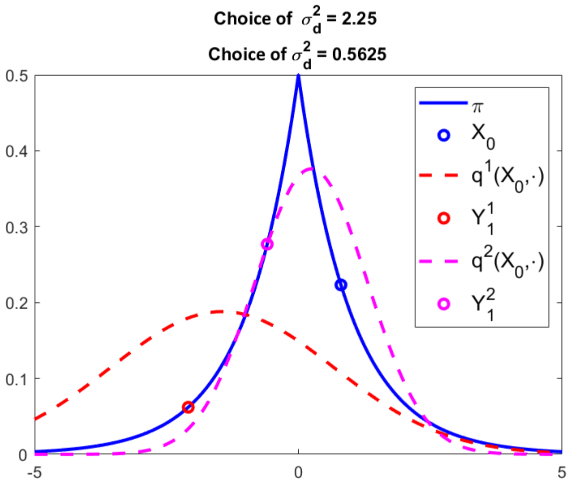
Step of MALA.

- Proposals $(Y_n^d)_n$ given by discretization:

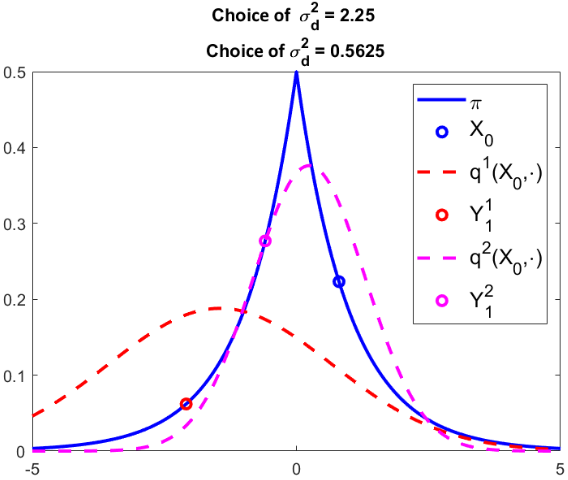
$$Y_{n+1}^d = X_n^d - \sigma_d^2 \nabla V\left(X_n^d\right) + \sqrt{2\sigma_d^2}\, Z_{n+1}^d ,$$

  with Z_{n+1}^d sampled from $\mathrm{N}(0, \mathrm{Id}_d)$.
- Biased gaussian proposal towards high density.
- Tune σ_d^2 w.r.t. the target π.

How to optimize σ_d^2 w.r.t. dimension d and target π?



How to optimize σ_d^2 w.r.t. dimension d and target π?



Before answering, explain what an "optimal choice" is.

The optimal scaling problem

How to tune the step-size of the proposal?

In the case of RWM and MALA: finding a "good" choise of σ_d^2 and quantify the efficiency of the methods.

Objectives:

1. approximate the behavior in high dimension with a simpler object,
2. find an optimization criterion for σ_d^2,
3. compare different methods.

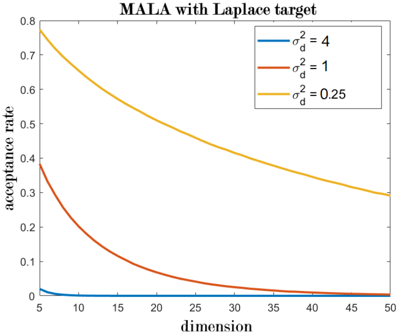
An asymptotical frame

- From target π over $\mathbb{R}$, consider a sequence $(\pi^d)_{d\geqslant 1}$

$$\pi^d(x^d) = \prod_{i=1}^{d} \pi\left(x_i^d\right) \ .$$

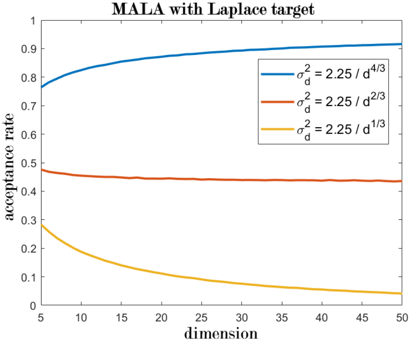
- Consider the sequence of Markov chains $\{(X_n^d)_{n\geqslant 0} \mid d \geqslant 1\}$, started at stationnarity, over the same probabilistic space.
- Goal: study the limit of $\mathbb{P}(\mathrm{A}_1^d)$ when $d \to +\infty$.

- $\mathrm{A}_1^d = \left\{ U_1 \leqslant \alpha(X_0^d, Y_1^d) \right\} .$



- Problem: if σ_d^2 constant w.r.t. d, $\lim_{d\to+\infty} \mathbb{P}(\mathrm{A}_1^d) = 0$.

First criterion for σ_d^2: scaling



Example of scaling in $\alpha = 2/3$.

- Choice: $\sigma_d^2 = \ell^2/d^\alpha$ with $\ell > 0$ and $\alpha > 0$.
- Goal: find α s.t.

$$a(\ell) = \lim_{d \to +\infty} \mathbb{P}(A_1^d) \in (0,1)$$

for any $\ell > 0$.

- We say that the method scales as $1/d^\alpha$.

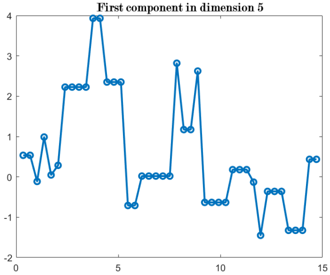
A scaling result

- Interpolate and scale X^d into $\mathbf{Y}^d$, with

$$\mathbf{Y}_t^d = (\lceil d^\alpha t \rceil - d^\alpha t)\, X^d_{\lfloor d^\alpha t \rfloor} + (d^\alpha t - \lfloor d^\alpha t \rfloor)\, X^d_{\lceil d^\alpha t \rceil}\,.$$

- Roberts et al. (1997): first component of $\mathbf{Y}^d$ converges in distribution towards $\mathbf{Y}$, solution of the SDE:

$$\mathrm{d}\mathbf{Y}_t = h(\ell) V'(\mathbf{Y}_t)\mathrm{d}t + \sqrt{2h(\ell)}\mathrm{d}\mathbf{B}_t\,.$$



Interpolation and scaling of the chain.

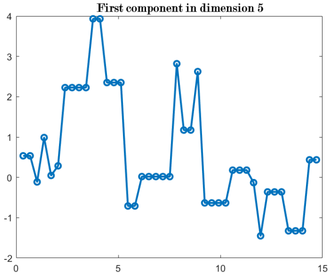
A scaling result

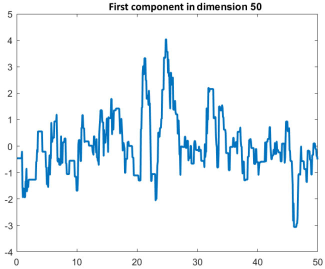
- Interpolate and scale X^d into $\mathbf{Y}^d$, with

$$\mathbf{Y}^d_t = \left(\lceil d^\alpha t \rceil - d^\alpha t\right) X^d_{\lfloor d^\alpha t \rfloor} + \left(d^\alpha t - \lfloor d^\alpha t \rfloor\right) X^d_{\lceil d^\alpha t \rceil} \, .$$

- Roberts et al. (1997): first component of $\mathbf{Y}^d$ converges in distribution towards $\mathbf{Y}$, solution of the SDE:

$$\mathrm{d}\mathbf{Y}_t = h(\ell) V'(\mathbf{Y}_t)\mathrm{d}t + \sqrt{2h(\ell)}\mathrm{d}\mathbf{B}_t \, .$$





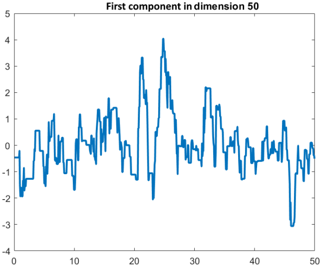
A scaling result

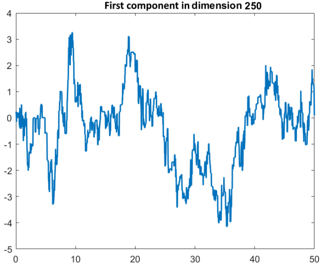
- Interpolate and scale X^d into $\mathbf{Y}^d$, with

$$\mathbf{Y}_t^d = (\lceil d^\alpha t \rceil - d^\alpha t)\, X^d_{\lfloor d^\alpha t \rfloor} + (d^\alpha t - \lfloor d^\alpha t \rfloor)\, X^d_{\lceil d^\alpha t \rceil}\,.$$

- Roberts et al. (1997): first component of $\mathbf{Y}^d$ converges in distribution towards $\mathbf{Y}$, solution of the SDE:

$$\mathrm{d}\mathbf{Y}_t = h(\ell) V'(\mathbf{Y}_t)\mathrm{d}t + \sqrt{2h(\ell)}\mathrm{d}\mathbf{B}_t\,.$$





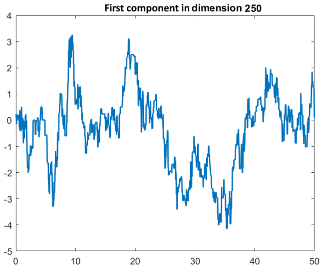
A scaling result

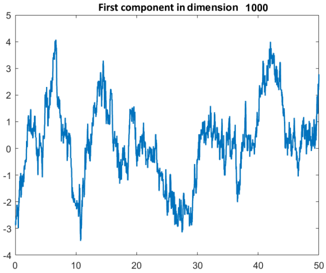
- Interpolate and scale X^d into $\mathbf{Y}^d$, with

$$\mathbf{Y}^d_t = (\lceil d^\alpha t \rceil - d^\alpha t) X^d_{\lfloor d^\alpha t \rfloor} + (d^\alpha t - \lfloor d^\alpha t \rfloor) X^d_{\lceil d^\alpha t \rceil} .$$

- Roberts et al. (1997): first component of $\mathbf{Y}^d$ converges in distribution towards $\mathbf{Y}$, solution of the SDE:

$$\mathrm{d}\mathbf{Y}_t = h(\ell) V'(\mathbf{Y}_t)\mathrm{d}t + \sqrt{2h(\ell)}\mathrm{d}\mathbf{B}_t .$$





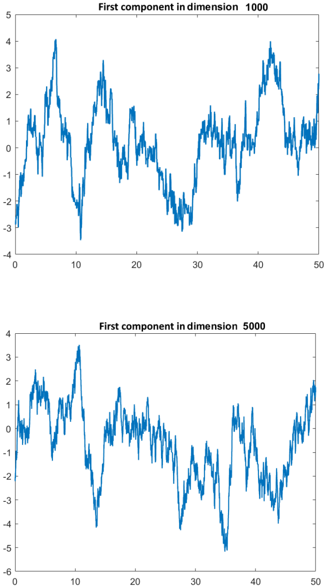
A scaling result

- Interpolate and scale X^d into $\mathbf{Y}^d$, with

$$\mathbf{Y}^d_t = (\lceil d^\alpha t \rceil - d^\alpha t)\, X^d_{\lfloor d^\alpha t \rfloor} + (d^\alpha t - \lfloor d^\alpha t \rfloor)\, X^d_{\lceil d^\alpha t \rceil} \,.$$

- Roberts et al. (1997): first component of $\mathbf{Y}^d$ converges in distribution towards $\mathbf{Y}$, solution of the SDE:

$$\mathrm{d}\mathbf{Y}_t = h(\ell) V'(\mathbf{Y}_t)\mathrm{d}t + \sqrt{2h(\ell)}\mathrm{d}\mathbf{B}_t \,.$$



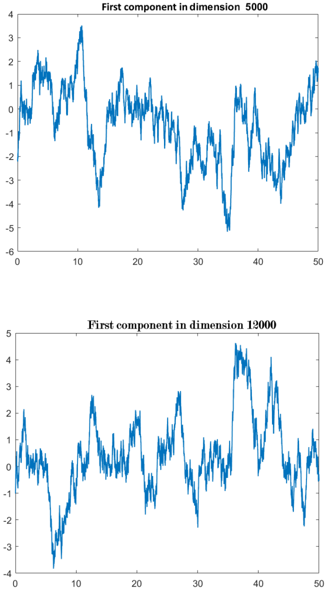
A scaling result

- Interpolate and scale X^d into $\mathbf{Y}^d$, with

$$\mathbf{Y}_t^d = (\lceil d^\alpha t \rceil - d^\alpha t) X^d_{\lfloor d^\alpha t \rfloor} + (d^\alpha t - \lfloor d^\alpha t \rfloor) X^d_{\lceil d^\alpha t \rceil} \, .$$

- Roberts et al. (1997): first component of $\mathbf{Y}^d$ converges in distribution towards $\mathbf{Y}$, solution of the SDE:

$$\mathrm{d}\mathbf{Y}_t = h(\ell) V'(\mathbf{Y}_t)\mathrm{d}t + \sqrt{2h(\ell)}\mathrm{d}\mathbf{B}_t \, .$$



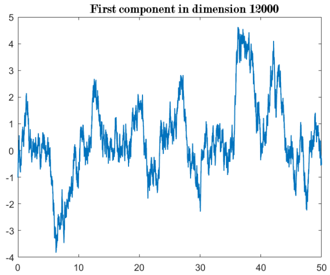
A scaling result

- Interpolate and scale X^d into $\mathbf{Y}^d$, with

$$\mathbf{Y}_t^d = (\lceil d^\alpha t \rceil - d^\alpha t)\, X^d_{\lfloor d^\alpha t \rfloor} + (d^\alpha t - \lfloor d^\alpha t \rfloor)\, X^d_{\lceil d^\alpha t \rceil}\,.$$

- Roberts et al. (1997): first component of $\mathbf{Y}^d$ converges in distribution towards $\mathbf{Y}$, solution of the SDE:

$$\mathrm{d}\mathbf{Y}_t = h(\ell) V'(\mathbf{Y}_t)\mathrm{d}t + \sqrt{2h(\ell)}\mathrm{d}\mathbf{B}_t\,.$$



How to tune the step-size of the proposal?

Objectives:

1. approximate the behavior in high dimension with a simpler object,
2. find an optimization criterion for σ_d^2,
3. compare different methods.

How to tune the step-size of the proposal?

Objectives:

1. approximate the behavior in high dimension with a simpler object, ✓ diffusive limit
2. find an optimization criterion for σ_d^2,
3. compare different methods.

- $\mathbf{Y}$ is solution of the SDE:

$$\mathrm{d}\mathbf{Y}_t = h(\ell)\,(\log\pi)'\,(\mathbf{Y}_t)\mathrm{d}t + \sqrt{2h(\ell)}\mathrm{d}\mathbf{B}_t\,. \tag{1}$$

- The diffusion speed h is the limit of the first order efficiency:

$$2h(\ell) = \lim_{d\to+\infty} d^\alpha \mathbb{E}\left[(X^d_{0,1} - X^d_{1,1})^2\right]\,.$$

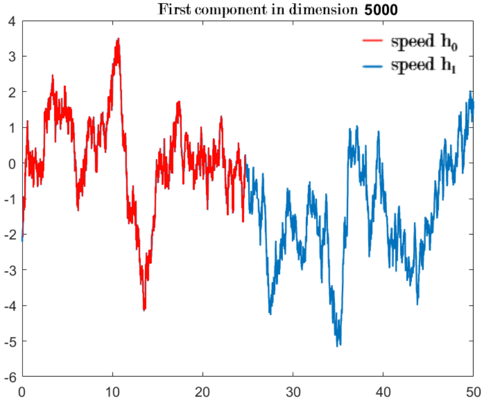
- If $\mathbf{Y}^{(1)}$ satisfies SDE (1) with $h = 1$, then

$$\left(\mathbf{Y}^{(1)}_{h(\ell)t}\right)_t \overset{\text{distr.}}{=} (\mathbf{Y}_t)_t\,.$$

Consequences of the diffusive limit

- If $\mathbf{Y}^{(1)}$ satisfies SDE (1) with $h = 1$, then

$$\left(\mathbf{Y}^{(1)}_{h(\ell)t}\right)_t \overset{\text{distr.}}{=} (\mathbf{Y}_t)_t \ .$$



$\neq$ diffusion speeds $h_0 < h_1$ $\rightsquigarrow \neq$ convergence speeds.

Consequences of the diffusive limit

- Roberts and Tweedie (1996) : $\mathbf{Y}^{(1)}$ is ergodic and π is an invariant distribution.

$\rightsquigarrow$ Maximizing h yields the best convergence.

- Scaling of $\mathbf{Y}^{d}$ in d^{α} :

$$\mathbf{Y}_t^d = (\lceil d^\alpha t \rceil - d^\alpha t)\, X^d_{\lfloor d^\alpha t \rfloor} + (d^\alpha t - \lfloor d^\alpha t \rfloor)\, X^d_{\lceil d^\alpha t \rceil} \,.$$

$\rightsquigarrow$ d^{α} is an estimate of the method's complexity.

How to tune the step-size of the proposal?

Objectives:

1. approximate the behavior in high dimension with a simpler object, ✓ diffusive limit,
2. find an optimization criterion for σ_d^2,
3. compare different methods.

How to tune the step-size of the proposal?

Objectives:

1. approximate the behavior in high dimension with a simpler object, ✓ diffusive limit,
2. find an optimization criterion for σ_d^2, ✓ $\sigma_d^2 = \ell^2/d^\alpha$ where ℓ maximises h,
3. compare different methods.

How to tune the step-size of the proposal?

Objectives:

1. approximate the behavior in high dimension with a simpler object, ✓ diffusive limit,
2. find an optimization criterion for σ_d^2, ✓ $\sigma_d^2 = \ell^2/d^\alpha$ where ℓ maximises h,
3. compare different methods, ✓ comparing the scaling value α $\rightsquigarrow$ better if α is small.

State of the art: comparing methods

Target	MALA	RWM
Smooth	Roberts and Rosenthal (1997) $\alpha = 1/3$, $\mathbf{a} = 0.574$	Roberts et al. (1997) $\alpha = 1$, $\mathbf{a} = 0.234$
Non-smooth	no general result	Durmus et al. (2017) $\alpha = 1$, $\mathbf{a} = 0.234$

- $\mathbf{a}$ is the optimal acceptance rate $\rightsquigarrow$ equivalent to optimize $\ell > 0$.

State of the art: comparing methods

Target	MALA	RWM
Smooth	Roberts and Rosenthal (1997) $\alpha = 1/3$, $\mathbf{a} = 0.574$	Roberts et al. (1997) $\alpha = 1$, $\mathbf{a} = 0.234$
Non-smooth	no general result	Durmus et al. (2017) $\alpha = 1$, $\mathbf{a} = 0.234$

- $\mathbf{a}$ is the optimal acceptance rate $\rightsquigarrow$ equivalent to optimize $\ell > 0$.
- Question: MALA for non-smooth targets?
- Special case $\pi : x \mapsto \mathrm{e}^{-|x|}/2$.

Main results

Optimal scaling of MALA with Laplace target

- Target $\pi : x \mapsto \mathrm{e}^{-|x|}/2$.
- Markov chain given by:

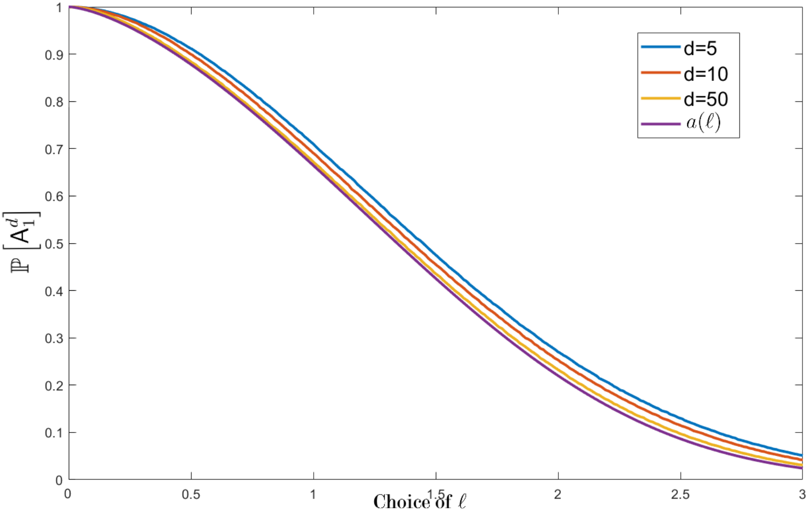
$$X_{n+1}^d = X_n^d + \left(-\frac{\ell^2}{d^\alpha}\,\mathrm{sgn}(X_n^d) + \sqrt{2}\frac{\ell}{d^{\alpha/2}} Z_{n+1}^d\right) \mathbb{1}_{\mathrm{A}_{n+1}^d}\,.$$

- Reminder: acceptance rate $\mathbb{P}(\mathrm{A}_1^d) = \mathbb{E}[\alpha(X_0^d, Y_1^d)]$.

Theorem (scaling value)

Let $\sigma_d^2 = \ell^2/d^\alpha$, with $\alpha = 2/3$, and Φ the $\mathrm{N}(0,1)$ c.d.f. . Then $\lim_{d\to+\infty} \mathbb{P}[\mathrm{A}_1^d] = a(\ell)$, where $a(\ell) = 2\Phi(-\ell^{3/2}/(3\pi^{1/2})^{1/2})$.

Example



Acceptance rate w.r.t. ℓ for MALA with Laplace target.

Optimal scaling of MALA with Laplace target

Theorem (diffusive limit and optimal scaling)

- $\alpha = 2/3$ and $(\mathbf{Y}_t^d)_{t \geqslant 0}$ given by

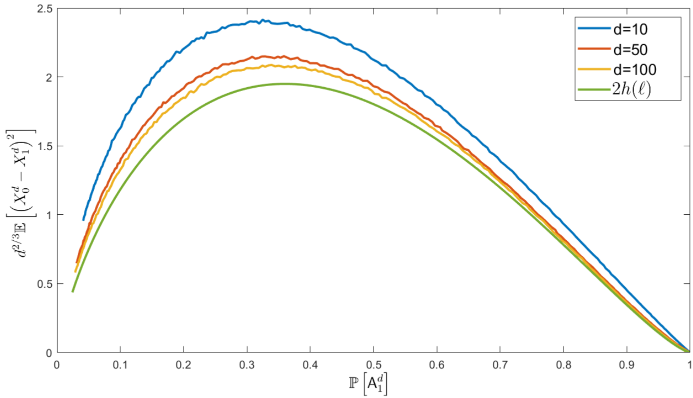
$$\mathbf{Y}_t^d = \left(\lceil d^{2/3} t \rceil - d^{2/3} t\right) X^d_{\lfloor d^{2/3} t \rfloor} + \left(d^{2/3} t - \lfloor d^{2/3} t \rfloor\right) X^d_{\lceil d^{2/3} t \rceil} .$$

- Let $(\mathbf{Y}_t)_{t \geqslant 0}$ be a solution of the SDE

$$\mathrm{d}\mathbf{Y}_t = h(\ell) \operatorname{sgn}(\mathbf{Y}_t) \mathrm{d}t + \sqrt{2h(\ell)} \mathrm{d}\mathbf{B}_t .$$

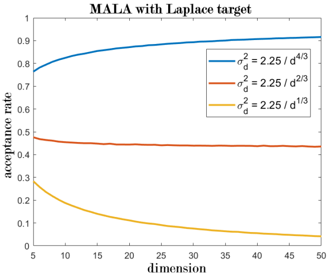
- Then $\{(\mathbf{Y}_{t,1}^d)_{t \geqslant 0} \mid d \in \mathbb{N}^*\}$ converges in distribution to $(\mathbf{Y}_t)_{t \geqslant 0}$.
- Moreover $h(\ell) = \ell^2 \Phi(-\ell^{3/2}/(3\pi^{1/2})^{1/2})$,
- h is maximized for ℓ s.t. $a(\ell) = \mathbf{a} = 0.360$.

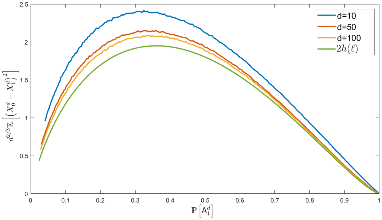
Illustration



First order efficiency times $d^{2/3}$ w.r.t. acceptance rate, for MALA with Laplace target.

Conclusion





Scaling results for MALA with Laplace target:

- New scaling in $\alpha = 2/3$.
- Weakly convergence of $\mathbf{Y}^d$ to a diffusive limit.
- New optimal acceptance rate $a(\ell) = 0.360$.

Perspective: generalize to more non-smooth densities e.g. continuous but not C^1.

Thank you for your time!

References

A. Durmus, S. Le Corff, E. Moulines, and G.O. Roberts. Optimal scaling of the random walk Metropolis algorithm under Lp mean differentiability. *Journal of Applied Probability*, 54(4):1233–1260, 2017. doi: 10.1017/jpr.2017.61.

W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. ISSN 00063444. URL http://www.jstor.org/stable/2334940.

N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, and A.H. Teller. Equations of state calculations by fast computing machine. *J. Chem. Phys.*, 21:1087–1091, 1953.

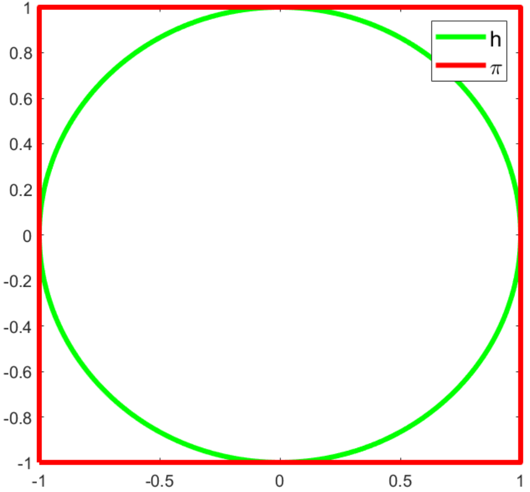
Gareth O. Roberts and Richard L. Tweedie. Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363, 12 1996. URL https://projecteuclid.org:443/euclid.bj/1178291835.

G.O. Roberts and J.S. Rosenthal. Optimal scaling of discrete approximations to Langevin diffusions. *J. R. Statist. Soc. B*, 60:255–268, 1997.

G.O. Roberts, A. Gelman, and W.R. Gilks. Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability*, 7(1):110–120, 1997.

Luke Tierney. Markov chains for exploring posterior distributions. *Ann. Statist.*, 22(4):1701–1728, 12 1994. doi: 10.1214/aos/1176325750. URL https://doi.org/10.1214/aos/1176325750.

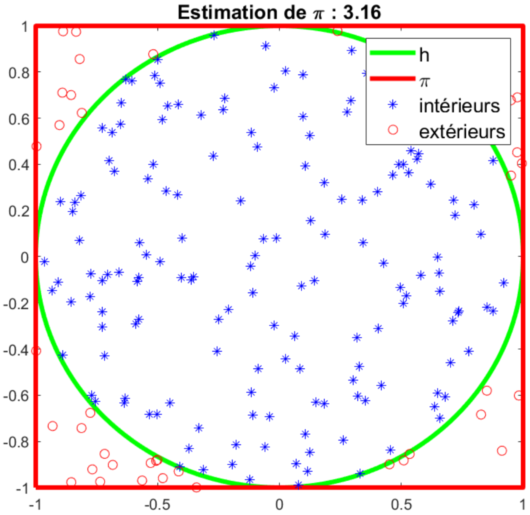
Monte Carlo method: example



- On veut estimer π.
- Alors $\pi = \mathrm{U}([-1,1]^2)$ et $h = \mathbb{1}_{\mathrm{B}(0,1)}$.
- $\int_{\mathbb{R}^2} h(x^2)\pi(x^2)\mathrm{d}x^2 = \pi/4$.

Continue presentation

Monte Carlo method: example

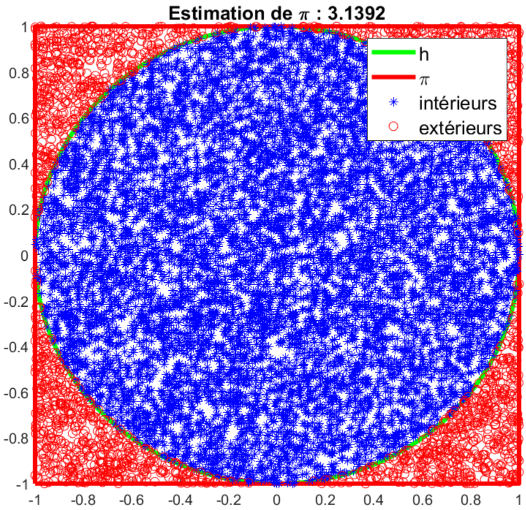


$n = 200$

- On veut estimer π.
- Alors $\pi = \mathrm{U}([-1,1]^2)$ et $h = \mathbb{1}_{\mathrm{B}(0,1)}$.
- $\int_{\mathbb{R}^2} h(x^2)\pi(x^2)\mathrm{d}x^2 = \pi/4$.

Continue presentation

Monte Carlo method: example



$n = 10000$

- On veut estimer π.
- Alors $\pi = \mathrm{U}([-1,1]^2)$ et $h = \mathbb{1}_{\mathrm{B}(0,1)}$.
- $\int_{\mathbb{R}^2} h(x^2)\pi(x^2)\mathrm{d}x^2 = \pi/4$.

Continue presentation