Adding some memory in stochastic algorihms

Pierre Monmarché

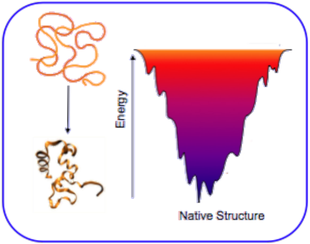
post-doc fellow at CERMICS

INRIA's Junior Seminar

High dimensional problems

Example: proteim folding, big data, path finding. . .

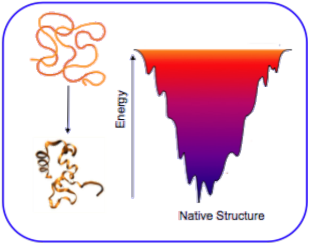
- Ω configuration space
- $x \in \Omega$ microscopic configuration
- $V(x)$: energy of x



High dimensional problems

Example: proteim folding, big data, path finding...

- Ω configuration space
- $x \in \Omega$ microscopic configuration
- $V(x)$: energy of x



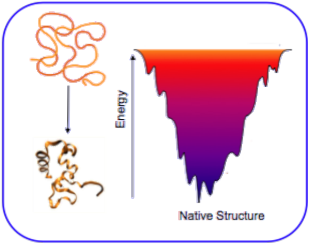
Two typical questions:

- Optimization: find x_0 such that $V(x_0) = \min_{\Omega} V$.

High dimensional problems

Example: proteim folding, big data, path finding...

- Ω configuration space
- $x \in \Omega$ microscopic configuration
- $V(x)$: energy of x



Two typical questions:

- Optimization: find x_0 such that $V(x_0) = \min_{\Omega} V$.
- Computing macroscopic quantities

$$\mathbb{E}\left(f(X)\right) = \frac{\int_\Omega f(x) e^{-\beta V(x)} \mathrm{d}x}{\int_\Omega e^{-\beta V(x)} \mathrm{d}x}$$

when X is random with Gibbs law at inverse temperature β.

Deterministic algorithms

Exhaustiveness: discretization (if necessary; Ω may already be discrete)

$$\Omega \simeq \{1, \ldots m\}^d$$

where $d = dim\Omega$ and $m =$ size of the mesh.

Deterministic algorithms

Exhaustiveness: discretization (if necessary; Ω may already be discrete)

$$\Omega \simeq \{1, \dots m\}^d$$

where $d = dim\Omega$ and $m =$ size of the mesh. Then

$$\frac{\int_\Omega f(x) e^{-\beta V(x)} \mathrm{d}x}{\int_\Omega e^{-\beta V(x)} \mathrm{d}x} \simeq \frac{\sum_\Omega f(x_i) e^{-\beta V(x_i)}}{\sum_\Omega e^{-\beta V(x_i)}},$$

and minimization by exhaustiveness.

Deterministic algorithms

Exhaustiveness: discretization (if necessary; Ω may already be discrete)

$$\Omega \simeq \{1,\dots m\}^d$$

where $d = dim\Omega$ and $m =$ size of the mesh. Then

$$\frac{\int_\Omega f(x)e^{-\beta V(x)}\mathrm{d}x}{\int_\Omega e^{-\beta V(x)}\mathrm{d}x} \simeq \frac{\sum_\Omega f(x_i)e^{-\beta V(x_i)}}{\sum_\Omega e^{-\beta V(x_i)}},$$

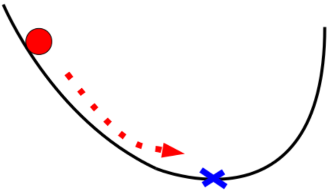
and minimization by exhaustiveness.

- $d = 3 \times 100$ amino acid in a protein, $m = 10 \Rightarrow$ Crazy.
- $30! \simeq 10^{32}$ different paths to connect 30 nodes $\Rightarrow$ Crazy.

Deterministic algorithms

Gradient descent: Start from $x_0 \in \Omega$, then

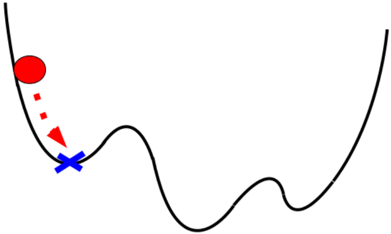
$$x'(t) \;=\; -\nabla V\left(x(t)\right).$$



Deterministic algorithms

Gradient descent: Start from $x_0 \in \Omega$, then

$$x'(t) = -\nabla V(x(t)) \,.$$

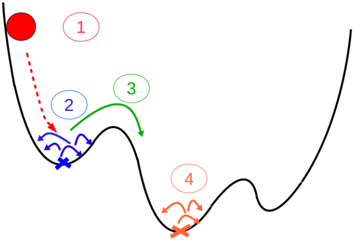


Problem: multi-modality.

Stochastic algorithms

Start from $x_0 \in \Omega$ and B a Brownian motion, then

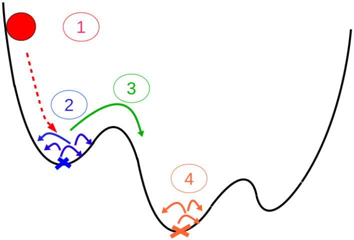
$$x'(t) \;=\; -\nabla V\left(x(t)\right) + \sqrt{2\beta^{-1}}\mathrm{d}B_t.$$



Stochastic algorithms

Start from $x_0 \in \Omega$ and B a Brownian motion, then

$$x'(t) \;=\; -\nabla V\left(x(t)\right) + \sqrt{2\beta^{-1}}\mathrm{d}B_t.$$



For large times t, $Law\left(x(t)\right) \simeq e^{-\beta V}$. The process is ergodic:

$$\frac{1}{t}\int_{s=0}^{t} f\left(x(s)\right)\mathrm{d}s \underset{t\to\infty}{\longrightarrow} \frac{\int_\Omega f(y)e^{-\beta V(y)}\mathrm{d}y}{\int_\Omega e^{-\beta V(y)}\mathrm{d}y}.$$

Markov Chain Monte Carlo (MCM) algorithms

Markov process/chain = no memory:

$$x'(t) \;=\; -\nabla V\left(x(t)\right) + \sqrt{2\beta^{-1}}\mathrm{d}B_t,$$

or, with a standard Gaussian variable G and a stepsize δ,

$$X_{n+1} \;=\; X_n - \delta\nabla V\left(X_n\right) + \sqrt{2\delta\beta^{-1}}G.$$

The past, $(X_k)_{k<n}$, is not needed (= is not used).

Markov Chain Monte Carlo (MCM) algorithms

Markov process/chain = no memory:

$$x'(t) \;=\; -\nabla V\left(x(t)\right) + \sqrt{2\beta^{-1}}\mathrm{d}B_t,$$

or, with a standard Gaussian variable G and a stepsize δ,

$$X_{n+1} \;=\; X_n - \delta\nabla V\left(X_n\right) + \sqrt{2\delta\beta^{-1}}G.$$

The past, $(X_k)_{k<n}$, is not needed (= is not used). Same for Metropolis-Hastings, etc.

Markov Chain Monte Carlo (MCM) algorithms

Markov process/chain = no memory:

$$x'(t) \;=\; -\nabla V\left(x(t)\right) + \sqrt{2\beta^{-1}}\mathrm{d}B_t,$$

or, with a standard Gaussian variable G and a stepsize δ,

$$X_{n+1} \;=\; X_n - \delta \nabla V\left(X_n\right) + \sqrt{2\delta\beta^{-1}}G.$$

The past, $(X_k)_{k<n}$, is not needed (= is not used). Same for Metropolis-Hastings, etc.

Amnesic exploration of Ω :

Inefficient !

(metastability)

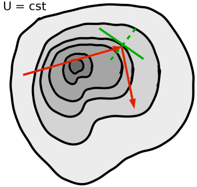
Instantaneous memory

Consider the velocity $y(t) = x'(t)$. Add some inertia.

Instantaneous memory

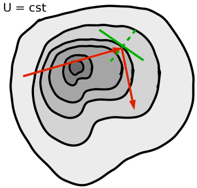
Consider the velocity $y(t) = x'(t)$. Add some inertia.

- Jump processes: y is piecewise constant, with random jumps (depending on the potential landscape)



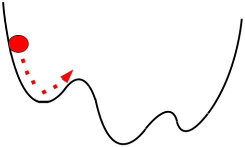
Instantaneous memory

Consider the velocity $y(t) = x'(t)$. Add some inertia.



- Jump processes: y is piecewise constant, with random jumps (depending on the potential landscape)

- Newton's law of motion:

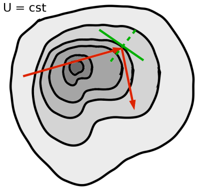


$$my'(t) = -\nabla V(x(t)) - \nu y(t) + \sqrt{2\beta^{-1}}\mathrm{d}B_t$$

(m = mass, ν = friction coefficient)

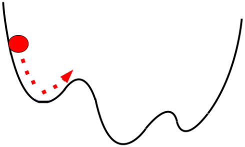
Instantaneous memory

Consider the velocity $y(t) = x'(t)$. Add some inertia.



- Jump processes: y is piecewise constant, with random jumps (depending on the potential landscape)

- Newton's law of motion:



$$my'(t) = -\nabla V(x(t)) - \nu y(t) + \sqrt{2\beta^{-1}}\mathrm{d}B_t$$

(m = mass, ν = friction coefficient)

So that in either case, again,

$$\frac{1}{t}\int_{s=0}^{t} f(x(s))\,\mathrm{d}s \underset{t\to\infty}{\longrightarrow} \frac{1}{\int_\Omega e^{-\beta V(y)}\mathrm{d}y}\int_\Omega f(y)e^{-\beta V(y)}\mathrm{d}y.$$

Degenerated Markov process

Example of the Langevin dynamics:

$$\begin{aligned} x'(t) &= y(t) \\ y'(t) &= -\nabla V\left(x(t)\right) - y(t) + \sqrt{2\beta^{-1}}\mathrm{d}B_t. \end{aligned}$$

Degenerated Markov process

Example of the Langevin dynamics:

$$\begin{aligned} x'(t) &= y(t) \\ y'(t) &= -\nabla V\left(x(t)\right) - y(t) + \sqrt{2\beta^{-1}}\mathrm{d}B_t. \end{aligned}$$

Then $Law\left(x(t), y(t)\right) = \rho_t(u,v)\mathrm{d}u\mathrm{d}v$ solves

$$\partial_t \rho_t + v\nabla_u \rho_t = \nabla_v \cdot \left((V(u) + v)\rho \right) + \frac{1}{\beta}\Delta_v \rho$$

Degenerated Markov process

Example of the Langevin dynamics:

$$\begin{aligned} x'(t) &= y(t) \\ y'(t) &= -\nabla V(x(t)) - y(t) + \sqrt{2\beta^{-1}}\mathrm{d}B_t. \end{aligned}$$

Then $Law\,(x(t), y(t)) = \rho_t(u,v)\mathrm{d}u\mathrm{d}v$ solves

$$\partial_t \rho_t + v\nabla_u \rho_t = \nabla_v \cdot ((V(u)+v)\rho) + \frac{1}{\beta}\Delta_v \rho$$

"Degenerated" (linear) PDE : hypoelliptic, hypocoercive. Still, $\rho_t(u,v) \underset{t\to\infty}{\longrightarrow} e^{-V(u)-\frac{1}{2}|v|^2}$

Degenerated Markov process

Example of the Langevin dynamics:

$$\begin{aligned} x'(t) &= y(t) \\ y'(t) &= -\nabla V(x(t)) - y(t) + \sqrt{2\beta^{-1}} \mathrm{d}B_t. \end{aligned}$$

Then $Law\,(x(t), y(t)) = \rho_t(u,v)\mathrm{d}u\mathrm{d}v$ solves

$$\partial_t \rho_t + v\nabla_u \rho_t = \nabla_v \cdot ((V(u)+v)\rho) + \frac{1}{\beta}\Delta_v \rho$$

"Degenerated" (linear) PDE : hypoelliptic, hypocoercive. Still, $\rho_t(u,v) \underset{t\to\infty}{\longrightarrow} e^{-V(u)-\frac{1}{2}|v|^2}$, but theory more difficult than

$$x'(t) = -\nabla V(x(t)) + \sqrt{2\beta^{-1}} \mathrm{d}B_t$$

(overdamped Langevin) for which $\rho_t(u)$ solves

$$\partial_t \rho_t = \nabla_u \cdot (V(u)\rho) + \frac{1}{\beta}\Delta_u \rho.$$

Adaptative Biasing Force (ABF) method

- Microscopic configuration: $x = (x_1, \ldots, x_d) \in \Omega = (\mathbb{T})^d$
- Reaction coordinates: (x_1, x_2)
- Free energy: $A(x_1, x_2) \;=\; \frac{1}{\beta} \ln \int e^{-\beta V(x)} \mathrm{d}x_3 \ldots \mathrm{d}x_d$

If X random with law $e^{-\beta V}$, then (X_1, X_2) with law $e^{-\beta A}$.

Adaptative Biasing Force (ABF) method

- Microscopic configuration: $x = (x_1, \ldots, x_d) \in \Omega = (\mathbb{T})^d$
- Reaction coordinates: (x_1, x_2)
- Free energy: $A(x_1, x_2) \; = \; \frac{1}{\beta} \ln \int e^{-\beta V(x)} \mathrm{d}x_3 \ldots \mathrm{d}x_d$

If X random with law $e^{-\beta V}$, then (X_1, X_2) with law $e^{-\beta A}$.

Biased overdamped Langevin: sample $e^{-\beta(V-A)}$ with

$$x'(t) \; = \; -\nabla V\left(x(t)\right) - \nabla A\left(x_1(t), x_2(t)\right) + \sqrt{2\beta^{-1}}\mathrm{d}B_t.$$

If X random with law $e^{-\beta(V-A)}$, then
(X_1, X_2) is uniform on $\mathbb{T}^2$

Adaptative Biasing Force (ABF) method

- Microscopic configuration: $x = (x_1, \dots, x_d) \in \Omega = (\mathbb{T})^d$
- Reaction coordinates: (x_1, x_2)
- Free energy: $A(x_1, x_2) \;=\; \frac{1}{\beta} \ln \int e^{-\beta V(x)} \mathrm{d}x_3 \dots \mathrm{d}x_d$

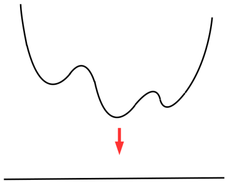
If X random with law $e^{-\beta V}$, then (X_1, X_2) with law $e^{-\beta A}$.

Biased overdamped Langevin: sample $e^{-\beta(V-A)}$ with

$$x'(t) \;=\; -\nabla V\left(x(t)\right) - \nabla A\left(x_1(t), x_2(t)\right) + \sqrt{2\beta^{-1}}\mathrm{d}B_t.$$

If X random with law $e^{-\beta(V-A)}$, then (X_1, X_2) is uniform on $\mathbb{T}^2$:

the metastability/multimodality disappeared !



Adaptative Biasing Force (ABF) method

Un-biasing:

$$\frac{\int_\Omega f e^{-\beta V}}{\int_\Omega e^{-\beta V}} = \frac{\int_\Omega \left(f e^{-\beta A}\right) e^{-\beta(V-A)}}{\int_\Omega \left(e^{-\beta A}\right) e^{-\beta(V-A)}}$$
$$= \lim_{t\to\infty} \frac{\int_0^t f\left(X(s)\right) e^{-\beta A(X_1(s),X_2(s))} \mathrm{d}s}{\int_0^t e^{-\beta A(X_1(s),X_2(s))} \mathrm{d}s}$$

with a biased X.

Adaptative Biasing Force (ABF) method

Un-biasing:

$$\frac{\int_\Omega f e^{-\beta V}}{\int_\Omega e^{-\beta V}} = \frac{\int_\Omega \left(f e^{-\beta A}\right) e^{-\beta(V-A)}}{\int_\Omega \left(e^{-\beta A}\right) e^{-\beta(V-A)}}$$
$$= \lim_{t\to\infty} \frac{\int_0^t f\left(X(s)\right) e^{-\beta A(X_1(s),X_2(s))} \mathrm{d}s}{\int_0^t e^{-\beta A(X_1(s),X_2(s))} \mathrm{d}s}$$

with a biased X.

- Benefit: the biased X converges faster to its equilibrium

Adaptative Biasing Force (ABF) method

Un-biasing:

$$\frac{\int_\Omega f e^{-\beta V}}{\int_\Omega e^{-\beta V}} = \frac{\int_\Omega \left(f e^{-\beta A}\right) e^{-\beta(V-A)}}{\int_\Omega \left(e^{-\beta A}\right) e^{-\beta(V-A)}}$$

$$= \lim_{t\to\infty} \frac{\int_0^t f\left(X(s)\right) e^{-\beta A(X_1(s),X_2(s))} \mathrm{d}s}{\int_0^t e^{-\beta A(X_1(s),X_2(s))} \mathrm{d}s}$$

with a biased X.

- Benefit: the biased X converges faster to its equilibrium
- Drawback: ... A is unknown (precisely our aim in some cases)

Low dimensional, long-term memory

Solution: learn the bias ∇A on the fly, with:

$$\nabla A(x_1, x_2) = \frac{\int \nabla V(x) e^{-\beta V(x)} \mathrm{d}x_3 \ldots \mathrm{d}x_d}{\int e^{-\beta V(x)} \mathrm{d}x_3 \ldots \mathrm{d}x_d}$$

Low dimensional, long-term memory

Solution: learn the bias ∇A on the fly, with:

$$
\begin{aligned}
\nabla A(x_1, x_2) &= \frac{\int \nabla V(x) e^{-\beta V(x)} \mathrm{d}x_3 \dots \mathrm{d}x_d}{\int e^{-\beta V(x)} \mathrm{d}x_3 \dots \mathrm{d}x_d} \\
&\simeq \frac{\sum_{s=1}^{t} \nabla V\left(X(s)\right) 1_{(X_1(s), X_2(s)) = (x_1, x_2)}}{\sum_{s=1}^{t} 1_{(X_1(s), X_2(s)) = (x_1, x_2)}}
\end{aligned}
$$

Low dimensional, long-term memory

Solution: learn the bias ∇A on the fly, with:

$$\begin{aligned}\nabla A(x_1,x_2) &= \frac{\int \nabla V(x) e^{-\beta V(x)} \mathsf{d}x_3 \dots \mathsf{d}x_d}{\int e^{-\beta V(x)} \mathsf{d}x_3 \dots \mathsf{d}x_d} \\ &\simeq \frac{\sum_{s=1}^t \nabla V\left(X(s)\right) 1_{(X_1(s),X_2(s))=(x_1,x_2)}}{\sum_{s=1}^t 1_{(X_1(s),X_2(s))=(x_1,x_2)}}\end{aligned}$$

We keep a long-term memory: for all $(x_1,x_2) \in \{0, \frac{1}{m}, \frac{2}{m}, \dots, 1\}^2$,

$$\sum_{s=1}^t 1_{(X_1(s),X_2(s))=(x_1,x_2)} = \sharp\{\text{transit through cell } (x_1,x_2)\}$$

My current job

More reaction coordinates: $(x_1, \ldots, x_k)$. Memory of size m^k: not good.

My current job

More reaction coordinates: $(x_1, \ldots, x_k)$. Memory of size m^k: not good.

The idea: A is a function of k variables. It can be approached (known greedy algorithms) by a sum of tensor products of functions of 1 variable:

$$\begin{aligned} A(x_1, \ldots, x_k) \simeq\ & r_{1,1}(x_1) r_{1,2}(x_2) \ldots r_{1,k}(x_k) + \ldots \\ & + r_{n,1}(x_1) r_{n,2}(x_2) \ldots r_{n,k}(x_k) \end{aligned}$$

My current job

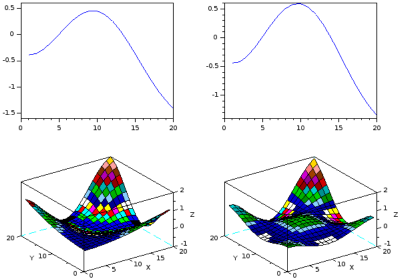
More reaction coordinates: $(x_1, \ldots, x_k)$. Memory of size m^k: not good.

The idea: A is a function of k variables. It can be approached (known greedy algorithms) by a sum of tensor products of functions of 1 variable:

$$\begin{aligned} A(x_1, \ldots, x_k) &\simeq r_{1,1}(x_1) r_{1,2}(x_2) \ldots r_{1,k}(x_k) \ + \ \ldots \\ & \quad + r_{n,1}(x_1) r_{n,2}(x_2) \ldots r_{n,k}(x_k) \end{aligned}$$

Example:

$$\begin{aligned} G(x,y) &= x^2 \cos(y) + y^2 \cos(x) \\ &\simeq r_1(x) r_2(y) \end{aligned}$$



My current job

After tensorization, memory of size $m \times k \times n$ with

- $m =$ size of the mesh
- $k =$ number of reaction coordinates
- $n =$ number of tensor products (not necessarily fixed)

My current job

After tensorization, memory of size $m \times k \times n$ with

- $m =$ size of the mesh
- $k =$ number of reaction coordinates
- $n =$ number of tensor products (not necessarily fixed)

New difficulty:

How should the bias be updated ? A tensor product is not "local"...

My current job

After tensorization, memory of size $m \times k \times n$ with

- $m =$ size of the mesh
- $k =$ number of reaction coordinates
- $n =$ number of tensor products (not necessarily fixed)

New difficulty:

How should the bias be updated ? A tensor product is not "local"...

Propspects:

Stastitics on the reaction coordinates (which one are really useful ? Does it depend on the region of Ω ? Which one are correlated ?)

My current job

After tensorization, memory of size $m \times k \times n$ with

- $m =$ size of the mesh
- $k =$ number of reaction coordinates
- $n =$ number of tensor products (not necessarily fixed)

New difficulty:

How should the bias be updated ? A tensor product is not "local"...

Propspects:

Stastitics on the reaction coordinates (which one are really useful ? Does it depend on the region of Ω ? Which one are correlated ?)

Thank you for your attention !