SSC Gold Medal Address
Conférence SSC Médaille d'Or

The Magic of / La Magie de Monte Carlo

Jeffrey S. Rosenthal

University of Toronto / l'Université de Toronto

www.probability.ca

(27 May/mai 2014)

My research can be divided into two parts :
Ma recherche peut se diviser en deux parties :

1. Theoretical foundations of Markov chain Monte Carlo (MCMC) algorithms. 1. Fondations théoriques des algorithmes de Monte Carlo par chaînes de Markov (MCMC).

2. Interdisciplinary applications of statistics. 2. Applications interdisciplinaires de la statistique.

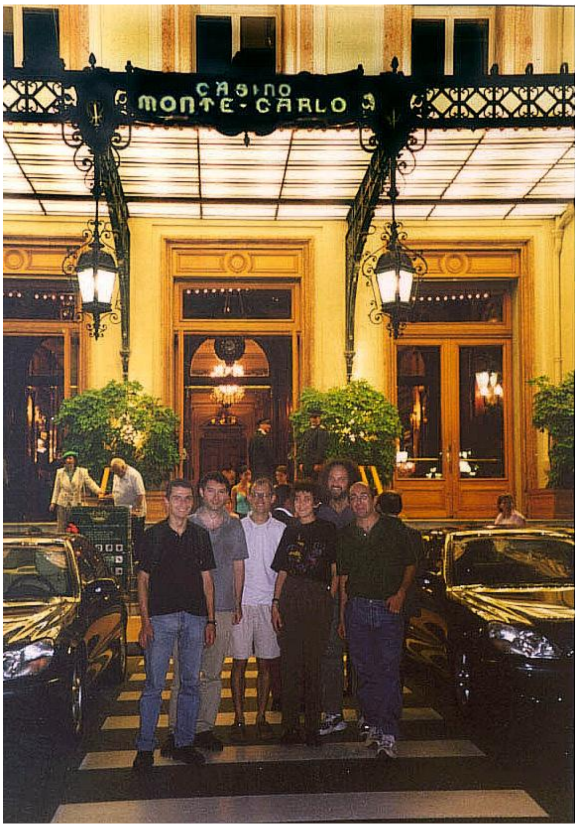
I will summarise item 2 in the companion article "Interdisciplinary Sojourns", to appear in the *Canadian Journal of Statistics*. J'offrirai un resumé de numéro 2 dans un article pour la Revue Canadienne de Statistique. Thanks to / Grâce à David Stephens.

Today I will discuss item 1. Aujourd'hui je parlerai de numéro 1.

By the way, I did the slides' French translations myself. Mon dieu, comme les oeufs sont chauds ce matin !

The Magic of / La Magie de Monte Carlo ?

Nice Place for a Conference ! Agréable pour un congrès !



Monte Carlo in a nutshell : To sample is to know. L'essence de Monte Carlo : l'échantillon, c'est la connaissance.

Suppose π is an important (but complicated) probability distribution, e.g. a Bayesian posterior. Soit π une distribution de probabilité importante (mais compliquée), p.e. une postérieure bayésienne.

If $X_1, X_2, \ldots, X_M$ is a sample from π, we can use it to : Nous pouvons utiliser un échantillon $X_1, X_2, \ldots, X_M$ de π pour :

- See a picture of π : histogram, density estimate. Voir une image de π : histogramme, estimé de la densité.
- Estimate the mean of π / Estimer la moyenne de π : $\frac{1}{M}\sum_{i=1}^{M} X_i$.
- Or the mean of any function h of π / Ou la moyenne de n'importe quelle fonction h de π : $\mathbf{E}_\pi(h) \approx \frac{1}{M}\sum_{i=1}^{M} h(X_i)$.
- Or the probability of any event A / Ou la probabilité de n'importe quel événement A : $\mathbf{P}_\pi(A) \approx \frac{1}{M}\sum_{i=1}^{M} \mathbf{1}(X_i \in A)$.
- To sample is to know! L'échantillon, c'est la connaissance! (5/22)

Extremely popular! Extrêmement populaire!

Google hits/résultats "Markov chain Monte Carlo" = 1,820,000.

Widely used in / Largement utilisé pour :

- Bayesian Inference / l'inférence bayésienne
- Medical Research / la recherche médicale
- Statistical Genetics / la génétique statistique
- Chemical Physics / la physique chimique
- Computer Science / la science informatique
- Mathematical Finance / la finance mathématique
- Engineering / l'ingénierie
- etc.

So, important to understand! Alors, important à comprendre!

But how can we sample ? Comment échantillonner ?

Use MCMC ! Utiliser des algorithmes MCMC !

e.g. Given a previous state X, propose a new state $Y \sim Q(X, \cdot)$ (symmetric). Étant donné un ancien état X, proposer un nouvel état $Y \sim Q(X, \cdot)$ (symétrique).

Then if $\pi(Y) > \pi(X)$, accept the new state. Si $\pi(Y) > \pi(X)$, accepter le nouvel état.

Otherwise, accept it only with probability $\pi(Y) / \pi(X)$. Sinon, accepter seulement avec probabilité $\pi(Y) / \pi(X)$.

Then watch the magic ! Puis regarder la magie ! [rwm.java]

Empirical distribution (black) converges to target (blue).
La distribution empirique (noir) converge vers la distribution cible (bleu).

MCMC works ! MCMC marche bien !

Example : Suppose we have n particles on a region, which depend on each other. Exemple : Supposons qu'il y a n particules dans une région, qui dépendent les uns des autres.

For example, suppose the probability of a configuration is proportional to e^{-H}, where H is an energy function, e.g. /
Par exemple, supposons que la probabilité d'une configuration est proportionelle à e^{-H}, où H est une fonction d'énergie, p.e.

$$H = \sum_{i<j} A\Big|(x_i, y_i) - (x_j, y_j)\Big| + \sum_{i<j} \frac{B}{\Big|(x_i, y_i) - (x_j, y_j)\Big|} + \sum_i C\, x_i$$

$A = B = C = 0$: independent / indépendant (Poisson).

A large : particles like to be close together / les particules veulent être proches.

B large : particles like to be far apart / loin.

C large : particles like to be towards the left / à la gauche.

Sample and see ! Voir l'échantillon ! [pointproc.java]

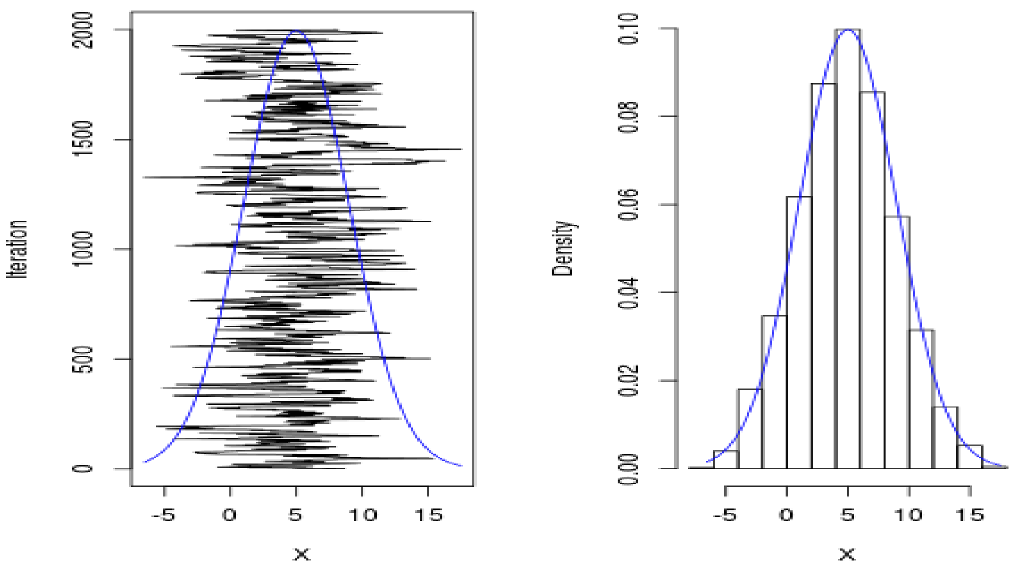
Continuous Density ? / Densité continue ?

Let's try it ! Essayons-le !

(Left : trace plot, with "time" moving upwards. Right : histogram.)

(Gauche : les valeurs X ; temps vers le haut. Droite : histogramme.)

[Rnorm]   It works well ! Ça marche bien !



Research questions ? Questions de recherche ?

Convergence bounds / Bornes de convergence : How quickly does MCMC converge ? À quelle vitesse converge le MCMC ?

Want to prove that black and blue are within 0.01 (say) after some specific number n_* of iterations. Nous voudrions prouver que le noir et le bleu sont à distance moins de 0.01 (disons) après un nombre spécifique n_* d'itérations. Difficult ! Difficile !

Some progress, esp. by "coupling" two different algorithm versions together. Quelque progrès, surtout avec le « couplage » de deux versions de l'algorithme. (R., JASA 1995, Stat & Comput. 1996).

e.g. I proved that a certain 20-dimensional Bayesian posterior example (with real data) was within 0.01 after just $n_* = 140$ iterations – good ! p.e. j'ai établi qu'un certain exemple bayésien en 20 dimensions (avec des données réelles) était entre 0.01 après $n_* = 140$ itérations – bon !

But generally too difficult. Généralement, c'est trop difficile.

Simpler research question / Question de recherche plus simple :

Is the convergence of black to blue exponentially fast ("geometrically ergodic")? Est-ce que la convergence du noir au bleu est exponentielle (« géométriquement ergodique »)?
i.e. dist $\leq C\,\rho^n$, for some / pour quelque $\rho < 1$.

This can make a huge difference! Ça peut changer beaucoup!

e.g. $\pi = \mathrm{Exp}(1)$, $Q(\cdot) = \mathrm{Exp}(0.01)$: geometric, converges within 0.01 after $n_* = 459$ iterations / géométrique, $n_* = 459$.

e.g. $\pi = \mathrm{Exp}(1)$, $Q(\cdot) = \mathrm{Exp}(5)$: not geometric / pas géométrique, $n_* > 4{,}000{,}000$. (Roberts & R., MCAP, 2011)

So, geometric ergodicity is widely studied and quite useful. L'ergodicité géométrique est largement étudiée et bien utile.

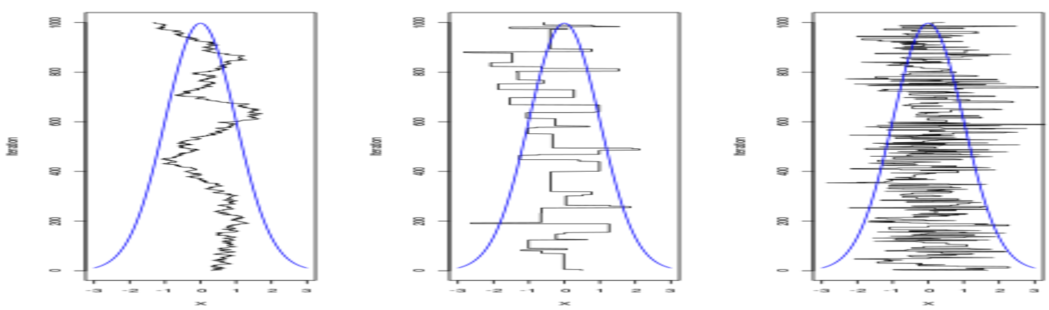
But can theory help to improve the algorithms? Mais est-ce que la théorie peut aider à améliorer les algorithmes?

Let's see! Voyons!

Question : What proposal works best? Quoi proposer?

Example/Exemple : Target/cible $\pi = N(0,1)$. Suppose we propose from / Supposons que nous proposons de $Q(x,\cdot) = N(x,\sigma^2)$.

How to choose σ? Comment choisir σ? [Rscale]



$\sigma = 0.1$?	$\sigma = 25$?	$\sigma = 2.38$?
too small / trop petit!	too big / trop grand!	just right / parfait!
A.R./T.d'A. $= 0.962$	0.052	0.441

So, want "moderate" σ, and "moderate" acceptance rate.
Préférer des σ et des taux d'acceptation « modérés ».

I call this the "Goldilocks Principle" : the best proposals are not too big, and not too small, but "just right".

≪ Le principe de Boucle d'or ≫ : les meilleures propositions sont ni trop grandes, ni trop petites, mais ≪ parfaites ≫.

Can theory tell us more ? Yes ! La théorie peut expliquer plus ? Oui !

Under "certain assumptions", as $d \to \infty$, if we speed up time and shrink space, then the Metropolis algorithm converges weakly to a diffusion. Avec ≪ certaines hypothèses ≫, quand $d \to \infty$, avec le temps plus vite et l'espace plus petit, l'algorithme Metropolis converge vers une diffusion.

(Just like how simple random walk, sped up and shrunk down, converges to Brownian motion. Comme la convergence des marches aléatoires vers le mouvement brownien.)

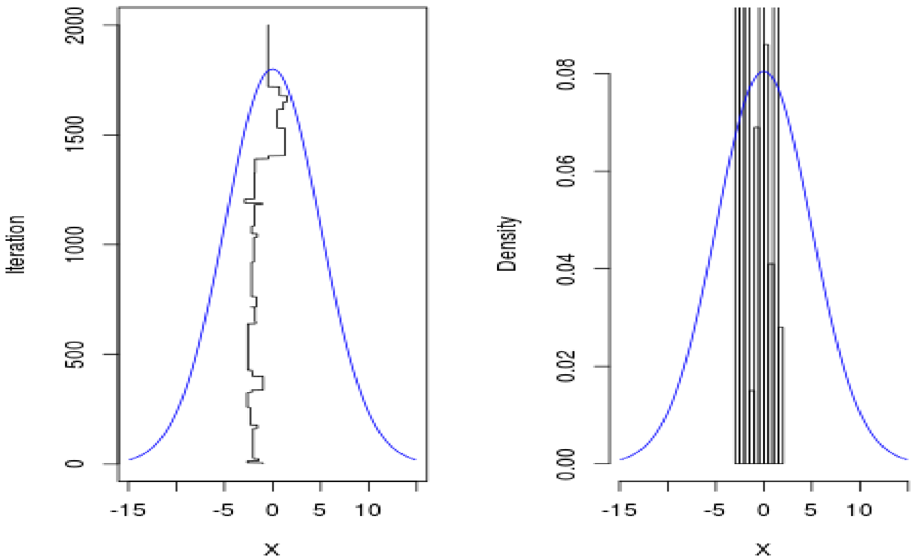
The diffusion is fastest (i.e. "optimised") when the proposals are "just right". La diffusion est la plus vite quand les propositions sont ≪ parfaites ≫. [Roberts, Gelman, Gilks, AAP 1997 ; Roberts & R., JRSSB 1998, Stat Sci 2001 ; Bédard, AAP 2007 ; Bédard & R., CJS 2008 ; . . .]

Optimal acceptance rate / Taux d'acceptation optimal : 0.234.

Clear, simple rule. Good ! Suggestion claire et simple. Bon ! But is "0.234" the whole story ? Est-ce que ≪ 0.234 ≫ nous dit tout ?

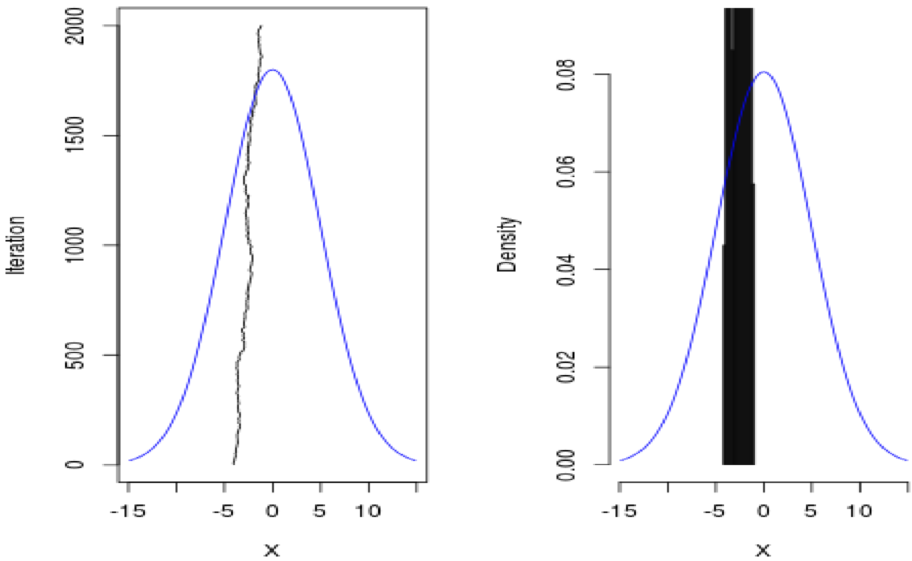
20-Dimensional Example / Exemple en 20 dimensions

Suppose π is a density on $\mathbf{R}^{20}$. Soit π une densité sur $\mathbf{R}^{20}$. Proposal covariance / Covariance de proposition $\Sigma = I_{20}$? [Rtwenty]



Acceptance rate / taux d'acceptation = 0.017. Too small / Trop petit ! Need smaller Σ plus petit !

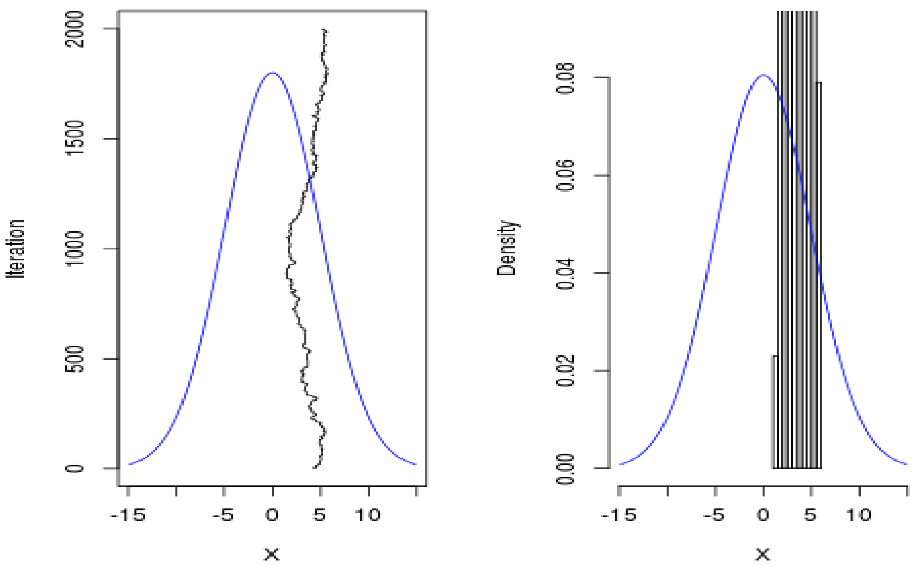
Second try / Deuxième effort : $\Sigma = 0.001 * I_{20}$. [Rtwenty]



Acceptance rate / taux d'acceptation = 0.652.

Too big / Trop grand ! Need bigger Σ plus grand !

Third try / Troisième effort : $\Sigma = 0.02 * I_{20}$. [Rtwenty]

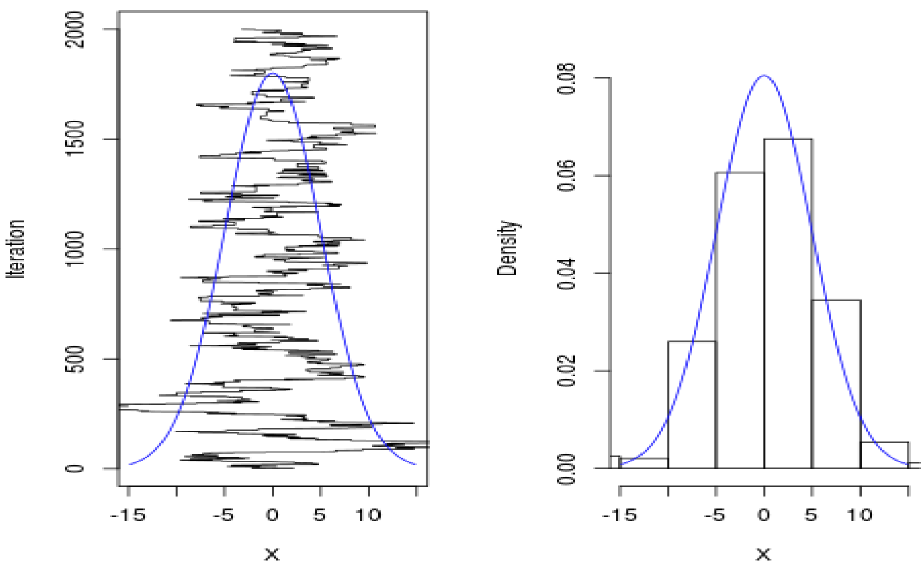


Acceptance rate / taux d'acceptation ≈ 0.234.

"Just right". « Parfait ».

So, why such poor performance ? Alors, pourquoi est-ce que ça ne marche pas bien ?

Fourth try / Quatrième effort : $\Sigma = \Sigma_{opt} := \frac{(2.38)^2}{20} \Sigma_\pi$, where Σ_π is the covariance of π / où Σ_π est la covariance de π. [Rtwenty]



Acceptance rate / taux d'acceptation ≈ 0.234 still / toujours. Performance now better / meilleure performance. THM : Σ_{opt} is optimal under "certain assumptions" / est optimal avec « certaines hypothèses » [Roberts & R., Stat Sci 2001].

Adaptive MCMC adaptatif

Know that optimal proposal covariance is / la covariance de proposition optimale est : $\frac{(2.38)^2}{\text{dim}}\,\Sigma_\pi$

What if Σ_π is unknown ? Mais si Σ_π n'est pas connu ?

Can replace Σ_π with the empirical estimate Σ_n from the run so far. Remplacer Σ_π par l'estimé empirique Σ_n de l'algorithme jusqu'à date. [Haario et al., 2001 ; Roberts & R., JCGS 2009]

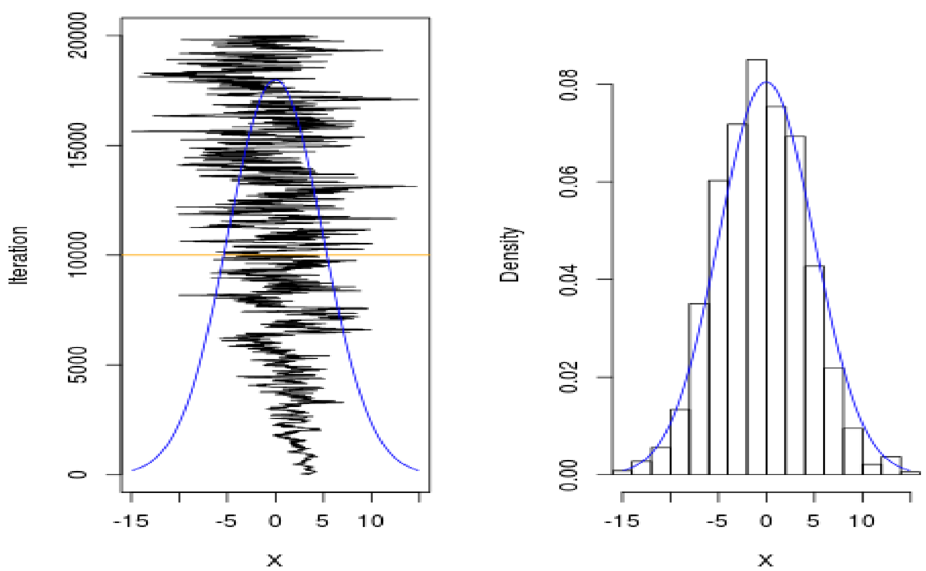
- Σ_n approximates / se rapproche de Σ_π. Good / bon.

- But it destroys the Markov property – bad. Ça ne respecte pas la proprieté markovienne – dommage.

- Does it work well ? Ça marche bien ?

- Does it converge ? Ça converge ?

How well does it work ? Ça marche bien ? [R20adapt ?]



In 20 dimensions, takes 10,000 iterations, then finds good proposal covariances and starts mixing well. Good ! En vingt dimensions, ça prend 10,000 itérations, puis ça trouve de bonnes covariances de proposition et ça marche bien. Bon !

Adaptive MCMC theory : does the chain converge ? La théorie de l'adaptation : est-ce que la chaîne converge ? [adapt.java]

Difficult – no longer Markovian ! Difficile – pas markovien !

Still converges under "certain assumptions". Ça converge sous « certaines hypothèses ». [Roberts & R., JAP 2007, JCGS 2009 ; Haario, Saksman, Tamminen, Vihola, Andrieu, Moulines, Robert, Fort, Atchadé, Craiu, Bai, Kohn, Giordani, Nott, . . .]

e.g. "Diminishing Adaptation" (easy/facile) and/et "Containment" (hard/difficile) [Roberts & R., JAP 2007]. Alternatives ?

- Alternative #1 : adapt only within a bounded region, and prove Containment using probabilistic arguments. Adapter seulement dans une région bornée, et établir Containment avec des arguments probabilistes. [Craiu, R., et al., submitted/soumis].

- Alternative #2 : cease adapting once the adaption has "stabilised". Finir l'adaptation quand elle est « stabilisée ». [J. Yang & R., in progress / en progrès]. May make adaption more generally applicable. Va peut-être rendre plus applicable l'adaptation.

Summary / Resumé

- Monte Carlo and MCMC are very widely used, to sample from distributions π. Monte Carlo et les algorithmes MCMC sont très largement utilisés, pour faire des échantillons des distributions π.

- MCMC theory has played a crucial supporting role. La théorie a bien aidé. For example / Par exemple :

- Convergence bounds / Bornes de convergence.

- Optimal acceptance rate / Taux d'acceptation optimal. (Goldilocks Principle / Boucle d'or)

- Optimal covariance optimale : $\Sigma_{opt} = \frac{(2.38)^2}{\dim} \Sigma_\pi$.

- Adaption : replace/remplacer Σ_π by/par Σ_n. (Works well ! Ça marche bien ! Still converges ? Ça converge toujours ?)

- Lots more to study ! Beaucoup plus à étudier !

All my papers, applets / mes articles et applets : www.probability.ca