Statistics and learning
Monte Carlo Markov Chains (methods)

Emmanuel Rachelson and Matthieu Vignes

ISAE SupAero

22\textsuperscript{nd} March 2013
Monte Carlo computation

Why, what?

An old experiment that conceived the idea of Monte Carlo methods is that of “Buffon’s needle”: you throw a $l$-length needle on a flat surface made of parallel lines with spacing $D (> l)$. Under ideal conditions, $P(\text{needle crosses one of the lines}) = \frac{2l}{\pi D}$. → Estimation of $\pi$ thanks to a large number of thrown needles:

$$\pi = \lim_{n \to \infty} \frac{2l}{P_n D},$$

where $P_n$ is the proportion of crosses in $n$ such throws.
Monte Carlo computation

Why, what?

- An old experiment that conceived the idea of Monte Carlo methods is that of “Buffon’s needle”: you throw a \( l \)-length needle on a flat surface made of parallel lines with spacing \( D > l \). Under ideal conditions, \( P(\text{needle crosses one of the lines}) = \frac{2l}{\pi D} \). → Estimation of \( \pi \) thanks to a large number of thrown needles:

\[
\pi = \lim_{n \to \infty} \frac{2l}{P_n D},
\]

where \( P_n \) is the proportion of crosses in \( n \) such throws.

- Basic concept here is that of simulating random processes in order to help evaluate some quantities of interest.
Monte Carlo computation

Why, what?

- An old experiment that conceived the idea of Monte Carlo methods is that of “Buffon’s needle”: you throw a $l$-length needle on a flat surface made of parallel lines with spacing $D (> l)$. Under ideal conditions, $P(\text{needle crosses one of the lines}) = \frac{2l}{\pi D}$. → Estimation of $\pi$ thanks to a large number of thrown needles:

$$\pi = \lim_{n \to \infty} \frac{2l}{P_n D},$$

where $P_n$ is the proportion of crosses in $n$ such throws.

- Basic concept here is that of simulating random processes in order to help evaluate some quantities of interest.

- First intensive use during WW II in order to make a good use of computing facilities (ENIAC): neutron random diffusion for atomic bomb design and the estimation of eigenvalues in the Schrödinger equation. Intensively developed by (statistical) physicists.
Monte Carlo computation

Why, what?

▶ An old experiment that conceived the idea of Monte Carlo methods is that of “Buffon’s needle”: you throw a $l$-length needle on a flat surface made of parallel lines with spacing $D (> l)$. Under ideal conditions, $P(\text{needle crosses one of the lines}) = \frac{2l}{\pi D}$. → Estimation of $\pi$ thanks to a large number of thrown needles:

$$\pi = \lim_{n \to \infty} \frac{2l}{P_n D},$$

where $P_n$ is the proportion of crosses in $n$ such throws.

▶ Basic concept here is that of simulating random processes in order to help evaluate some quantities of interest.

▶ First intensive use during WW II in order to make a good use of computing facilities (ENIAC): neutron random diffusion for atomic bomb design and the estimation of eigenvalues in the Schrödinger equation. Intensively developped by (statistical) physicists.

▶ main interest when no closed form of solutions is tractable.
Typical problems

1. Integral computation

\[ I = \int h(x)f(x)dx, \]

can be assimilated to a \( E_f[h] \) if \( f \) is a density distribution. To be written

\[ \int h(x) \frac{f(x)}{g(x)} g(x)dx = E_g[hf/g], \]

if \( f \) was not a density distribution and \( \text{Supp}(f) \subset \text{Supp}(g) \).
Typical problems

1. Integral computation

\[ I = \int h(x)f(x) \, dx, \]

can be assimilated to a \( E_f[h] \) if \( f \) is a density distribution. To be written
\[ \int h(x) \frac{f(x)}{g(x)} g(x) \, dx = E_g[h f / g], \]
if \( f \) was not a density distribution and \( \text{Supp}(f) \subset \text{Supp}(g) \).

2. Optimisation

\[ \max_{x \in X} f(x) \text{ or } \arg\max_{x \in X} f(x) \]

(min can replace max)
Need of Monte Carlo techniques: integration

- Essential part in many scientific problems: computation of

\[ I = \int_D f(x) \, dx. \]
Need of Monte Carlo techniques: integration

- Essential part in many scientific problems: computation of

\[ I = \int_D f(x) \, dx. \]

- If we can draw iid random samples from \( D \), we can compute

\[ \hat{I}_n = \frac{\sum_j (f(x^{(j)}))}{n} \] and LLN says: \( \lim_n \hat{I}_n = I \) with probability 1 and CLT give convergence rate:

\[ \sqrt{n}(\hat{I}_n - I) \to \mathcal{N}(0, \sigma^2), \]

where \( \sigma^2 = \text{var}(g(x)) \).
Need of Monte Carlo techniques: integration

- Essential part in many scientific problems: computation of
  \[ I = \int_D f(x)dx. \]

- If we can draw iid random samples from \( D \), we can compute
  \[ \hat{I}_n = \frac{1}{n} \sum_j f(x^{(j)}) \]
  and LLN says: \( \lim_n \hat{I}_n = I \) with probability 1
  and CLT give convergence rate:
  \[ \sqrt{n}(\hat{I}_n - I) \rightarrow \mathcal{N}(O, \sigma^2), \]
  where \( \sigma^2 = \text{var}(g(x)) \).

- In dimension 1, Riemann’s approximation give a \( \mathcal{O}(1/n) \) error rate.
  But deterministic methods fail when dimensionality increases.

- However, no free lunch theorem: in high-dimensional \( D \), (i) \( \sigma^2 \approx \text{how uniform } g \text{ is can be quite large} \) and (ii) issue to produce uniformly distributed sample in \( D \).

- Again, importance sampling theoretically solves this but the choice of sample distribution is a challenge.
Need of Monte Carlo techniques: integration

- Essential part in many scientific problems: computation of
  \[ I = \int_D f(x)dx. \]

- If we can draw iid random samples from \( D \), we can compute
  \[ \hat{I}_n = \sum_j (f(x^{(j)}))/n \]
  and LLN says: \( \lim_{n} \hat{I}_n = I \) with probability 1
  and CLT give convergence rate:
  \[ \sqrt{n}(\hat{I}_n - I) \to N(O, \sigma^2), \]
  where \( \sigma^2 = \text{var}(g(x)) \).

- In dimension 1, Riemann’s approximation give a \( O(1/n) \) error rate.
  But deterministic methods fail when dimensionality increases.

- However, no free lunch theorem: in high-dimensional \( D \), (i) \( \sigma^2 \approx \text{how uniform } g \text{ is can be quite large and (ii) issue to produce uniformly distributed sample in } D. \)
Need of Monte Carlo techniques: integration

- Essential part in many scientific problems: computation of

\[ I = \int_D f(x)dx. \]

- If we can draw iid random samples from \( D \), we can compute

\[ \hat{I}_n = \frac{\sum_j f(x^{(j)})}{n} \]

and LLN says: \( \lim_n \hat{I}_n = I \) with probability 1 and CLT give convergence rate:

\[ \sqrt{n}(\hat{I}_n - I) \to \mathcal{N}(0, \sigma^2), \]

where \( \sigma^2 = \text{var}(g(x)) \).

- In dimension 1, Riemann's approximation give a \( O(1/n) \) error rate. But deterministic methods fail when dimensionality increases.

- However, no free lunch theorem: in high-dimensional \( D \), (i) \( \sigma^2 \approx \) how uniform \( g \) is can be quite large and (ii) issue to produce uniformly distributed sample in \( D \).

- Again, **importance sampling** theoretically solves this but the choice of sample distribution is a challenge.
Integration

a classical Monte Carlo approach

If we try to evaluate $I = \int f(x)g(x)dx$, where $g$ is a density function:
$I = E_g[f]$ and then:
Integration

a classical Monte Carlo approach

If we try to evaluate $I = \int f(x)g(x)\,dx$, where $g$ is a density function:
$I = E_g[f]$ and then:

**classical Monte Carlo method**

$\hat{I}_n = 1/n \sum_{i=1}^{n} f(x_i)$, where $x_i \sim \mathcal{L}(f)$. 
Integration
a classical Monte Carlo approach

If we try to evaluate $I = \int f(x)g(x)\,dx$, where $g$ is a density function:

$I = E_g[f]$ and then:

**classical Monte Carlo method**

$\hat{I}_n = 1/n \sum_{i=1}^{n} f(x_i)$, where $x_i \sim \mathcal{L}(f)$.

Justified by LLN & CLT if $\int f^2g < \infty$. 
Integration

no density at first

If $f$ is not a density (or not a “good” one), then for any density $g$ whose support contains the support of $f$: $I = \int h(x) \frac{f(x)}{g(x)} g(x) dx = E_g[h f / g]$.

Similarly:
Integration

no density at first

If $f$ is not a density (or not a “good” one), then for any density $g$ whose support contains the support of $f$: $I = \int h(x) \frac{f(x)}{g(x)} g(x) dx = E_g[h f / g]$.

Similarly:

importance sampling Monte Carlo method

$$\hat{I}_n = \frac{1}{n} \sum_{i=1}^{n} h(y_i) f(y_i) / g(y_i), \text{ where } y_i \sim \mathcal{L}(g).$$
Integration

no density at first

If $f$ is not a density (or not a “good” one), then for any density $g$ whose support contains the support of $f$: $I = \int h(x) \frac{f(x)}{g(x)} g(x) dx = E_g[h f / g]$.

Similarly:

**importance sampling Monte Carlo method**

$$\hat{I}_n = \frac{1}{n} \sum_{i=1}^{n} h(y_i) f(y_i) / g(y_i), \text{ where } y_i \sim \mathcal{L}(g).$$

Same justification but $\int h^2 f^2 / g < \infty$. This is equivalent to $\text{Var}_g(I_n) = \text{Var}_g(1/n \sum_{i=1}^{n} h(Y_i) f(Y_i) / g(Y_i))$; $g$ must have an heavier tail than that of $f$. **Choice of $g$ ?**
Integration

no density at first

If $f$ is not a density (or not a “good” one), then for any density $g$ whose support contains the support of $f$: $I = \int h(x) \frac{f(x)}{g(x)} g(x) dx = E_g[hf/g]$.

Similarly:

importance sampling Monte Carlo method

$$\hat{I}_n = \frac{1}{n} \sum_{i=1}^{n} h(y_i)f(y_i)/g(y_i), \text{ where } y_i \sim \mathcal{L}(g).$$

Same justification but $\int h^2 f^2 / g < \infty$. This is equivalent to

$$\text{Var}_g(I_n) = \text{Var}_g(1/n \sum_{i=1}^{n} h(Y_i)f(Y_i)/g(Y_i)); g \text{ must have an heavier tail than that of } f.$$

**Choice of $g$?**

**Theorem (Rubinstein)**

_The density $g^*$ which minimises $\text{Var}(\hat{I}_n)$ (for all $n$) is_

$$g^*(x) = \frac{|h(x)|f(x)}{\int |h(y)|f(y)dy}.$$
Monte Carlo integration

- was this optimal $g^*$ really useful? Remember the denominator (if $h > 0$)?
Monte Carlo integration

- Was this optimal $g^*$ really useful? Remember the denominator (if $h > 0$)?
- In practice, we choose $g$ such that $\text{Var}(\hat{I}_n) < \infty$ and $|h| f / g \simeq C$. 
Monte Carlo integration

- was this optimal $g^*$ really useful? Remember the denominator (if $h > 0$)?
- In practice, we choose $g$ such that $\text{Var}(\hat{I}_n) < \infty$ and $|h|f/g \approx C$.
- If $g$ is known up to a constant, the estimator
  \[ \frac{1}{n} \sum_{i=1}^{n} \frac{h(y_i)f(y_i)}{g(y_i)} / \sum_{i=1}^{n} \frac{f(y_i)}{g(y_i)} \]
  can replace $I_n$. 

\[ \text{BUT the optimality of } g^* \text{ cannot give any clue on the variance of this estimator...} \]
Monte Carlo integration

- was this optimal $g^*$ really useful? Remember the denominator (if $h > 0$)?
- In practice, we choose $g$ such that $\text{Var}(\hat{I}_n) < \infty$ and $|h|f/g \simeq C$.
- If $g$ is known up to a constant, the estimator
  \[ \frac{1}{n} \sum_{i=1}^{n} h(y_i)f(y_i)/g(y_i)/\sum_{i=1}^{n} f(y_i)/g(y_i) \]
  can replace $I_n$.
- BUT the optimality of $g$ cannot give any clue on the variance of this estimator...
Monte Carlo for optimisation

- Goal: $\max_{x \in \mathcal{X}} f(x)$ or $\arg\max_{x \in \mathcal{X}} f(x)$. 

Very simple part 1: if $\mathcal{X}$ is bounded, take $(x_i) \sim U(\mathcal{X})$ and estimate the max by $\max_{i=1}^{n} f(x_i)$. If $\mathcal{X}$ is not bounded, use an adequate variable transformation.

Very simple part 2: if $f \geq 0$, estimate $\arg\max_{x \in \mathcal{X}} f(x)$ boils down to estimating the mode of the distribution with density $f/\int f$. Recipe becomes: take $(x_i) \sim L(f/\int f)$, the estimator is the mode of the histogram of the $x_i$'s. If $f \not\geq 0$, then work with $g(x) = \exp [f(x)]$ or $g(x) = \exp [f(x)](1+\exp [f(x)])$.

In the latter case, the problem is the computation of the normalisation constant!

1. Newton-Raphson like methods: MCNR (MC approximation of score integrals and Hessian matrices) or StochasticApproximationNR.
2. EM-like approximations: MCEM or StochasticApproximationMC.
Monte Carlo for optimisation

- Goal: $\max_{x \in \mathcal{X}} f(x)$ or $\arg\max_{x \in \mathcal{X}} f(x)$.
- Very simple part 1: if $\mathcal{X}$ is bounded, take $(x_i) \sim \mathcal{U}(\mathcal{X})$ and estimate the max by $\max_{i=1\ldots n} f(x_i)$. If $\mathcal{X}$ is not bounded, use an adequate variable transformation.

- Very simple part 2: if $f \geq 0$, estimate $\arg\max_{x \in \mathcal{X}} f(x)$ boils down to estimating the mode of the distribution with density $f / \int f$. Recipe becomes: take $(x_i) \sim \mathcal{L}(f / \int f)$, the estimator is the mode of the histogram of the $x_i$'s. If $f \not\geq 0$, then work with $g(x) = \exp[f(x)]$ or $g(x) = \exp[f(x)](1+\exp[f(x)])$.

- In the latter case, the problem is the computation of the normalisation constant $!$.

- 1. Newton-Raphson like methods: MCNR (MC approximation of score integrals and Hessian matrices) or StochasticApproximationNR.
- 2. EM-like approximations: MCEM or StochasticApproximationMC.
Monte Carlo for optimisation

- Goal: $\max_{x \in \mathcal{X}} f(x)$ or $\arg\max_{x \in \mathcal{X}} f(x)$.

- Very simple part 1: if $\mathcal{X}$ is bounded, take $(x_i) \sim \mathcal{U}(\mathcal{X})$ and estimate the max by $\max_{i=1\ldots n} f(x_i)$. If $\mathcal{X}$ is not bounded, use an adequate variable transformation.

- Very simple part 2: if $f \geq 0$, estimate $\arg\max_{x \in \mathcal{X}} f(x)$ boils down to estimating the mode of the distribution with density $f/\int f$. Recipe becomes: take $(x_i) \sim \mathcal{L}(f/\int f)$, the estimator is the mode of the histogram of the $x_i$'s. If $f \not\geq 0$, then work with $g(x) = \exp[f(x)]$ or $g(x) = \frac{\exp[f(x)]}{1+\exp[f(x)]}$.
Monte Carlo for optimisation

- **Goal:** $\max_{x \in \mathcal{X}} f(x)$ or $\arg\max_{x \in \mathcal{X}} f(x)$.

- **Very simple part 1:** if $\mathcal{X}$ is bounded, take $(x_i) \sim \mathcal{U}(\mathcal{X})$ and estimate the max by $\max_{i=1\ldots n} f(x_i)$. If $\mathcal{X}$ is not bounded, use an adequate variable transformation.

- **Very simple part 2:** if $f \geq 0$, estimate $\arg\max_{x \in \mathcal{X}} f(x)$ boils down to estimating the mode of the distribution with density $f/\int f$. Recipe becomes: take $(x_i) \sim \mathcal{L}(f/\int f)$, the estimator is the mode of the histogram of the $x_i$'s. If $f \not\geq 0$, then work with $g(x) = \exp[f(x)]$ or $g(x) = \frac{\exp[f(x)]}{1+\exp[f(x)]}$.

- In the latter case, the problem is the computation of the normalisation constant!
Monte Carlo for optimisation

- Goal: $\max_{x \in \mathcal{X}} f(x)$ or $\arg\max_{x \in \mathcal{X}} f(x)$.

- Very simple part 1: if $\mathcal{X}$ is bounded, take $(x_i) \sim U(\mathcal{X})$ and estimate the max by $\max_{i=1 \ldots n} f(x_i)$. If $\mathcal{X}$ is not bounded, use an adequate variable transformation.

- Very simple part 2: if $f \geq 0$, estimate $\arg\max_{x \in \mathcal{X}} f(x)$ boils down to estimating the mode of the distribution with density $f/\int f$. Recipe becomes: take $(x_i) \sim \mathcal{L}(f/\int f)$, the estimator is the mode of the histogram of the $x_i$’s. If $f \not\geq 0$, then work with $g(x) = \exp [f(x)]$ or $g(x) = \frac{\exp [f(x)]}{1+\exp [f(x)]}$.

- In the latter case, the problem is the computation of the normalisation constant!
Monte Carlo for optimisation

- Goal: \( \max_{x \in \mathcal{X}} f(x) \) or \( \arg\max_{x \in \mathcal{X}} f(x) \).
- Very simple part 1: if \( \mathcal{X} \) is bounded, take \((x_i) \sim U(\mathcal{X})\) and estimate the max by \( \max_{i=1 \ldots n} f(x_i) \). If \( \mathcal{X} \) is not bounded, use an adequate variable transformation.
- Very simple part 2: if \( f \geq 0 \), estimate \( \arg\max_{x \in \mathcal{X}} f(x) \) boils down to estimating the mode of the distribution with density \( f / \int f \). Recipe becomes: take \((x_i) \sim \mathcal{L}(f / \int f)\), the estimator is the mode of the histogram of the \( x_i \)'s. If \( f \not\geq 0 \), then work with \( g(x) = \exp[f(x)] \) or \( g(x) = \frac{\exp[f(x)]}{1+\exp[f(x)]} \).
- In the latter case, the problem is the computation of the normalisation constant!
- 1. Newton-Raphson like methods: MCNR (MC approximation of score integrals and Hessian matrices) or StochasticApproximationNR.
Monte Carlo for optimisation

- **Goal:** \( \max_{x \in \mathcal{X}} f(x) \) or \( \arg\max_{x \in \mathcal{X}} f(x) \).

- **Very simple part 1:** if \( \mathcal{X} \) is bounded, take \( (x_i) \sim U(\mathcal{X}) \) and estimate the max by \( \max_{i=1\ldots n} f(x_i) \). If \( \mathcal{X} \) is not bounded, use an adequate variable transformation.

- **Very simple part 2:** if \( f \geq 0 \), estimate \( \arg\max_{x \in \mathcal{X}} f(x) \) boils down to estimating the mode of the distribution with density \( f/\int f \). Recipe becomes: take \( (x_i) \sim \mathcal{L}(f/\int f) \), the estimator is the mode of the histogram of the \( x_i \)'s. If \( f \not\geq 0 \), then work with \( g(x) = \exp[f(x)] \) or \( g(x) = \frac{\exp[f(x)]}{1+\exp[f(x)]} \).

- In the latter case, the problem is the computation of the normalisation constant!

- 1. Newton-Raphson like methods: MCNR (MC approximation of score integrals and Hessian matrices) or StochasticApproximationNR.
   2. EM-like approximations: MCEM or StochasticApproximationMC.
Monte Carlo vs numerical methods

- Numerical methods have lower computational cost in low dimension (integration) / would account for $f$ regularity, whilst MC methods won’t: no hypothesis on $f$ nor on $\mathcal{X}$ (optimisation).

- Advantage of MC methods 1 (integration): important support areas are given priority (whether the function varies a lot or its actual norm is great),

- Advantage of MC methods 2 (optimisation): local minima can be escaped and

- Advantage of MC methods 3: a straightforward extension to statistical inference (see next slide).

→ ideally, a method which efficiently combines the 2 points of view sounds much cleverer...
Monte Carlo vs numerical methods

- Numerical methods have lower computational cost in low dimension (integration) / would account for $f$ regularity, whilst MC methods won’t: no hypothesis on $f$ nor on $\mathcal{X}$ (optimisation).

- Advantage of MC methods 1 (integration): important support areas are given priority (whether the function varies a lot or its actual norm is great),
Monte Carlo vs numerical methods

- Numerical methods have lower computational cost in low dimension (integration) / would account for $f$ regularity, whilst MC methods won’t: no hypothesis on $f$ nor on $\mathcal{X}$ (optimisation).

- Advantage of MC methods 1 (integration): important support areas are given priority (whether the function varies a lot or its actual norm is great),

- Advantage of MC methods 2 (optimisation): local minima can be escaped and
Monte Carlo vs numerical methods

- Numerical methods have lower computational cost in low dimension (integration) / would account for $f$ regularity, whilst MC methods won’t: no hypothesis on $f$ nor on $\mathcal{X}$ (optimisation).

- Advantage of MC methods 1 (integration): important support areas are given priority (whether the function varies a lot or its actual norm is great),

- Advantage of MC methods 2 (optimisation): local minima can be escaped and

- Advantage of MC methods 3: a straightforward extension to statistical inference (see next slide).
Monte Carlo vs numerical methods

- Numerical methods have lower computational cost in low dimension (integration) / would account for $f$ regularity, whilst MC methods won’t: no hypothesis on $f$ nor on $x$ (optimisation).
- Advantage of MC methods 1 (integration): important support areas are given priority (whether the function varies a lot or its actual norm is great),
- Advantage of MC methods 2 (optimisation): local minima can be escaped and
- Advantage of MC methods 3: a straightforward extension to statistical inference (see next slide).
- Ideally, a method which efficiently combines the 2 points of view sounds much cleverer…
Monte Carlo and statistical inference

Integration

- Expectation computation
- Estimator precision estimation
- Bayesian analysis
- Mixture modelling or missing data treatment
Monte Carlo and statistical inference

Integration
- Expectation computation
- Estimator precision estimation
- Bayesian analysis
- Mixture modelling or missing data treatment

Optimisation
- Optimisation of some criterion,
- MLE,
- same last 2 points.
Monte Carlo and statistical inference
Bayesian framework

► Let \( x = (x_i)_{i=1...n} \) a sample with density known up to parameter \( \theta \in \Theta \).
Monte Carle and statistical inference
Bayesian framework

- Let \( x = (x_i)_{i=1}^n \) a sample with density known up to parameter \( \theta \in \Theta \).
- The **Bayesian approach** treats \( \theta \) as a rv with (prior) density \( \pi(\theta) \).

Bayes rule states that the posterior law is
\[
\pi(\theta | x) = \frac{\pi(\theta) f(x | \theta)}{\int \pi(\theta) f(x | \theta) \, d\theta}
\]
(note that often, the normalising constant is not tractable).

Main interests: (i) prior \( \pi \) permits to include prior knowledge on parameter and (ii) natural in some applications/modelling (Markov chains, mixture modelling, breakpoint detection . . . )
Monte Carlo and statistical inference

Bayesian framework

- Let $x = (x_i)_{i=1...n}$ a sample with density known up to parameter $\theta \in \Theta$.
- The **Bayesian approach** treats $\theta$ as a rv with (prior) density $\pi(\theta)$.
- We denote by $f(x|\theta)$ the density of $x$ conditional to $\theta$.
Monte Carlo and statistical inference
Bayesian framework

- Let $x = (x_i)_{i=1...n}$ a sample with density known up to parameter $\theta \in \Theta$.
- The **Bayesian approach** treats $\theta$ as a rv with (prior) density $\pi(\theta)$.
- We denote by $f(x|\theta)$ the density of $x$ conditional to $\theta$.
- Bayes rule states that the posterior law is $\pi(\theta|x) = \frac{\pi(\theta)f(x|\theta)}{\int \pi(\theta)f(x|\theta)d\theta}$ (note that often, the normalising constant is not tractable).

Main interests: (i) prior $\pi$ permits to include prior knowledge on parameter and (ii) natural in some applications/modelling (Markov chains, mixture modelling, breakpoint detection...)

E. Rachelson & M. Vignes (ISAE)  SAD  2013  11 / 19
Monte Carlo and statistical inference
Bayesian framework

- Let $x = (x_i)_{i=1...n}$ a sample with density known up to parameter $	heta \in \Theta$.
- The **Bayesian approach** treats $\theta$ as a rv with (prior) density $\pi(\theta)$.
- We denote by $f(x|\theta)$ the density of $x$ conditional to $\theta$.
- Bayes rule states that the posterior law is $\pi(\theta|x) = \frac{\pi(\theta)f(x|\theta)}{\int \pi(\theta)f(x|\theta)d\theta}$ (note that often, the normalising constant is not tractable).
- Main interests: (i) prior $\pi$ permits to include prior knowledge on parameter and (ii) natural in some applications/modelling (Markov chains, mixture modelling, breakpoint detection . . .)

E. Rachelson & M. Vignes (ISAE)
A Bayesian estimator $T(X)$ for $\theta$

in a nutshell

1. Choose a cost function $L(\theta, T(X))$ e.g. (i)
   $\mathbb{1}_\theta(T(X)) \Rightarrow T^*(x) = \arg\max_\theta \pi(\theta|x)$: optimisation problem or (ii)
   $\| T(X) - \theta \|^2 \Rightarrow T^*(x) = \int \theta \pi(\theta|x) d\theta$,
A Bayesian estimator $T(X)$ for $\theta$

in a nutshell

1. Choose a cost function $L(\theta, T(X))$ e.g. (i)
   \[ 1_{\theta}(T(X)) \Rightarrow T^*(x) = \arg\max_{\theta} \pi(\theta|x): \text{optimisation problem} \text{ or (ii)} \]
   \[ \| T(X) - \theta \|^2 \Rightarrow T^*(x) = \int \theta \pi(\theta|x)d\theta, \]
2. Derive the average risk: \[ R(T) = \int_X (\int_{\Theta} L(\theta, T(X)f(x|\theta)\pi(\theta)d\theta)dx, \]
A Bayesian estimator $T(X)$ for $\theta$

in a nutshell

1. Choose a cost function $L(\theta, T(X))$ e.g. (i) $1_{\theta}(T(X)) \Rightarrow T^*(x) = \arg\max_\theta \pi(\theta|x)$: optimisation problem or (ii) $\| T(X) - \theta \|^2 \Rightarrow T^*(x) = \int \theta \pi(\theta|x) d\theta$,

2. Derive the average risk: $R(T) = \int_X (\int_\Theta L(\theta, T(X)) f(x|\theta) \pi(\theta) d\theta) dx$,

3. Find the Bayesian estimator $T^* = \arg\min_T R(T)$,
A Bayesian estimator $T(X)$ for $\theta$

in a nutshell

1. Choose a cost function $L(\theta, T(X))$ e.g. (i)
   \[ 1_\theta(T(X) \Rightarrow T^*(x) = \arg\max_\theta \pi(\theta|x): \text{optimisation problem or (ii)} \]
   \[ \| T(X) - \theta \|^2 \Rightarrow T^*(x) = \int \theta \pi(\theta|x)d\theta, \]

2. Derive the average risk: $R(T) = \int_X (\int_\Theta L(\theta, T(X)f(x|\theta)\pi(\theta)d\theta)dx$,

3. Find the Bayesian estimator $T^* = \arg\min_T R(T)$,

4. The generalised Bayesian estimator is
   \[ T^*(x) = \arg\min_T \int_\Theta L(\theta, T(X)f(x|\theta)\pi(\theta)d\theta \text{ almost everywhere}. \]
MCMC methods

Why ? How ?

Why ?

Monte Carlo Markov Chain methods are used when the distribution under study cannot be simulated directly by usual techniques and/or when its density is known up to a constant.
MCMC methods

Why ? How ?

Why ?

Monte Carlo Markov Chain methods are used when the distribution under study cannot be simulated directly by usual techniques and/or when its density is known up to a constant.

How ?

An MCMC methods simulates a Markov chain \((X_i)_{i \geq 0}\) with transition kernel \(P\). The Markov chain converges in a sense to be precised towards the distribution of interest \(\pi\) (ergodicity property)
Ergodic theorem
for homogeneous Markov chains

Theorem

*Under certain conditions (recurrence and existence of an invariant distribution of for example), whatever the initial distribution \( \mu_0 \) for \( X_0 \), the distribution \( \mu_i \) is s.t.*

\[
\lim_{i \to \infty} \| \mu_i - \pi \| = 0 \quad \text{and} \\
1/n \sum_{i=0}^{n-1} h(X_k) \to E_\pi[h(X)] = \int h(x)\pi(x)dx \quad \text{a.s.}
\]
Ergodic theorem
for homogeneous Markov chains

Theorem

Under certain conditions (recurrence and existence of an invariant distribution ofr example), whatever the initial distribution $\mu_0$ for $X_0$, the distribution $\mu_i$ is s.t.

$$\lim_{i \to \infty} \| \mu_i - \pi \| = 0 \text{ and}$$

$$\frac{1}{n} \sum_{i=0}^{n-1} h(X_k) \to E_\pi[h(X)] = \int h(x)\pi(x)dx \text{ a.s.}$$

Remarks

- $(X_i)$’s are not independent but the ergodic theorem replace the LLN.
- Ergodic theorems exist under milder conditions and for inhomogeneous chains.
MCMC algorithms

Just like accept/reject methods or importance sampling, MCMC methods make use of an instrumental law. This instrumental law can be characterised by a transition kernel $q(\|)$ or by a conditional distribution.
MCMC algorithms

Just like accept/reject methods or importance sampling, MCMC methods make use of an instrumental law. This instrumental law can be characterised by a transition kernel $q(\parallel)$ or by a conditional distribution.

- Simulation and integration: Metropolis-Hastings algorithm or Gibbs sampling.
- Optimisation: simulated annealing.
Metropolis-Hastings algorithm

- **Initialisation:** $x_0$.
- **for each step $k \geq 0$:**
  1. Simulate a value $y_k$ from $Y_k \sim q(\cdot|x_k)$,
  2. Simulate a value $u_k$ from $U_k \sim U([0,1])$,
  3. Update

\[
x_{k+1} = \begin{cases} 
    y_k & \text{if } u_k \leq \rho(x_k, y_k) \\
    x_k & \text{otherwise},
  \end{cases}
\]

where $\rho(x, y) = \min \left(1, \frac{\pi(y)q(x|y)}{\pi(x)q(y|x)}\right)$.
Metropolis-Hastings algorithm

- Initialisation: \(x_0\).
- for each step \(k \geq 0\):
  1. Simulate a value \(y_k\) from \(Y_k \sim q(\cdot|x_k)\),
  2. Simulate a value \(u_k\) from \(U_k \sim U([0, 1])\),
  3. Update

\[
x_{k+1} = \begin{cases} 
y_k & \text{if } u_k \leq \rho(x_k, y_k) \\
x_k & \text{otherwise},
\end{cases}
\]

where \(\rho(x, y) = \min \left(1, \frac{\pi(y)q(x|y)}{\pi(x)q(y|x)} \right)\).

Note that only \(\pi(y)/\pi(x)\) and \(q(y|x)/q(x|y)\) ratios are needed, so no need to compute normalising constants!

Note also that while favourable move are always accepted, unfavourable move can be accepted (with a probability which decreases with the level of degradation).
Simulated annealing

Goal: minimise a real-valued function $f$. 

Clever practical modification: the objective function is changed over the iteration:

$$\pi(x) \propto \exp \left( -\frac{f(x)}{T_k} \right)$$

where $(T_k)$ is a non-increasing sequence of temperatures. In practice, the temperature is high in the first iterations to explore and avoid local minima and it then starts decreasing more or less rapidly towards 0.
Simulated annealing

Goal: minimise a real-valued function $f$.

Idea: Apply a Metropolis-Hastings algorithm to simulate the distribution
$\pi(x) \propto \exp(-f(x))$ and then estimate its mode(s).

Clever practical modification: the objective function is changed over the
iteration:
$\pi(x) \propto \exp(-f(x)/T_k)$, where $(T_k)$ is a non-increasing sequence of
temperatures.

In practice, the temperature is high in the first iterations to explore and
avoid local minima and it then starts decreasing more or less rapidly
towards 0.
Simulated annealing

Goal: minimise a real-valued function \( f \).

Idea: Apply a Metropolis-Hastings algorithm to simulate the distribution \( \pi(x) \propto \exp(-f(x)) \) and then estimate its mode(s).

Clever practical modification: the objective function is changed over the iteration:

\[
\pi(x) \propto \exp \left( -\frac{f(x)}{T_k} \right),
\]

where \((T_k)\) is a non-increasing sequence of temperatures. In practice, the temperature is high in the first iterations to explore and avoid local minima and it then starts decreasing more or less rapidly towards 0.
Simulated annealing algorithm

- Initialisation: $x_0$.
- for each step $k \geq 0$:
  1. Simulate a value $y_k$ from $Y_k \sim q(y|x_k)$,
  2. Simulate a value $u_k$ from $U_k \sim U([0, 1])$,
  3. Update
     \[
     x_{k+1} = \begin{cases} 
     y_k & \text{if } u_k \leq \rho(x_k, y_k) \\
     x_k & \text{otherwise,}
     \end{cases}
     \]
     where $\rho(x, y) = \min \left(1, \frac{e^{-f(y)/T_k} q(x|y)}{e^{-f(x)/T_k} q(y|x)}\right)$.
  4. Decrease temperature $T_k \rightarrow T_{k+1}$. 
This is over!
or almost

Was that clear enough? Too quick?

Some simple applications might help...