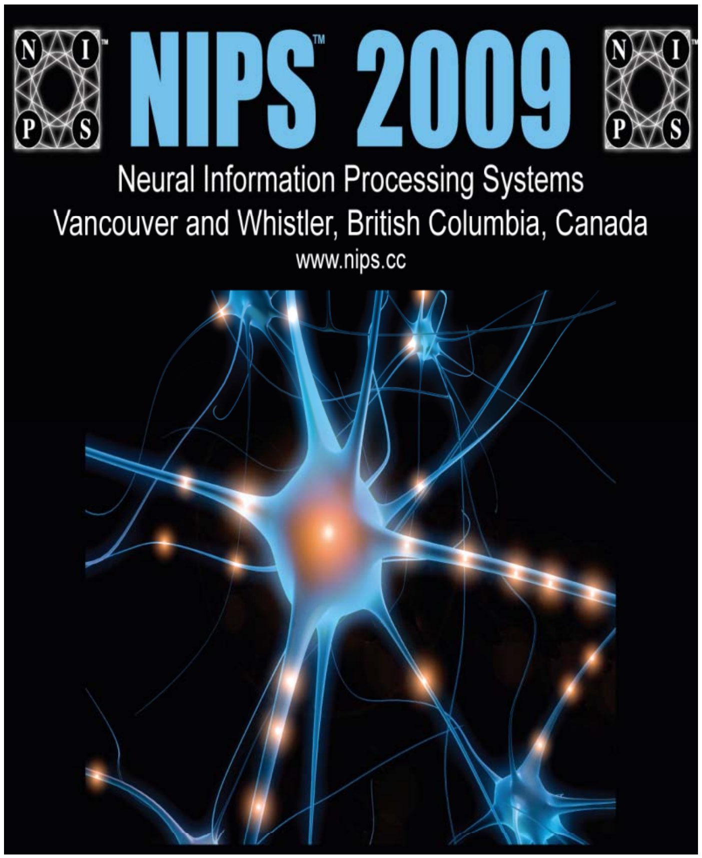


Sequential Monte Carlo

Nando de Freitas & Arnaud Doucet

University of British Columbia

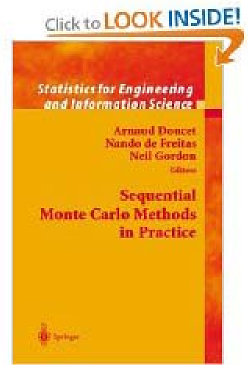
December 2009

Tutorial overview

- **Introduction** **Nando – 10min**
- **Part I** **Arnaud – 50min**
  - Monte Carlo
  - Sequential Monte Carlo
  - Theoretical convergence
  - Improved particle filters
  - Online Bayesian parameter estimation
  - Particle MCMC
  - Smoothing
  - Gradient based online parameter estimation
- **Break** **15min**
- **Part II** **NdF – 45 min**
  - Beyond state space models
  - Eigenvalue problems
  - Diffusion, protein folding & stochastic control
  - Time-varying Pitman-Yor Processes
  - SMC for static distributions
  - Boltzmann distributions & ABC

Tutorial overview

- **Introduction** **Nando – 10min**
- **Part I** **Arnaud – 50min**
  - Monte Carlo
  - Sequential Monte Carlo
  - Theoretical convergence
  - Improved particle filters
  - Online Bayesian parameter estimation
  - Particle MCMC
  - Smoothing
  - Gradient based online parameter estimation
- **Break** **15min**
- **Part II** **NdF – 45 min**
  - Beyond state space models
  - Eigenvalue problems
  - Diffusion, protein folding & stochastic control
  - Time-varying Pitman-Yor Processes
  - SMC for static distributions
  - Boltzmann distributions & ABC



20th century

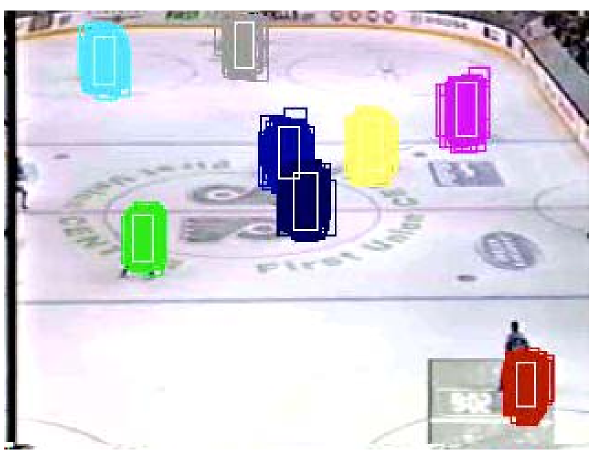
SMC in this community

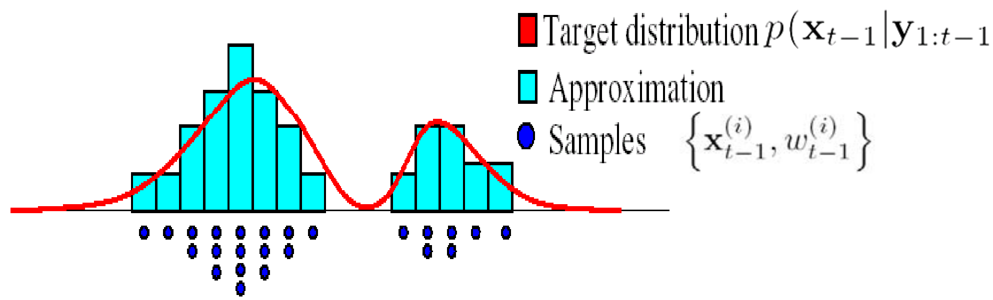
Many researchers in the NIPS community have contributed to the field of Sequential Monte Carlo over the last decade.

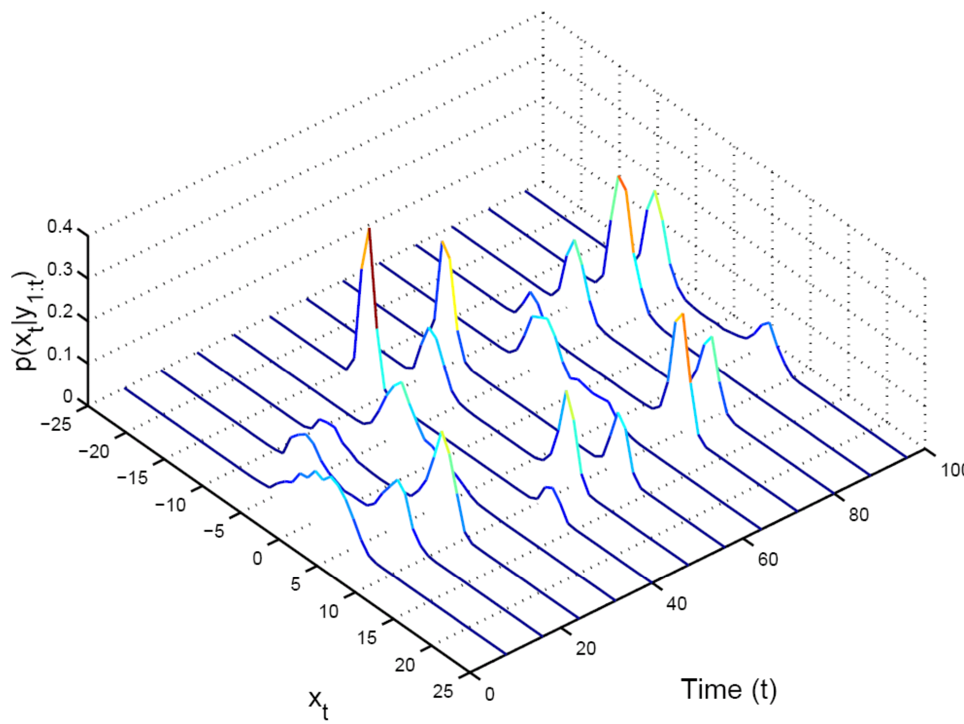
- *Michael Isard* and *Andrew Blake* popularized the method with their Condensation algorithm for image tracking.
- Soon after, *Daphne Koller*, *Stuart Russell*, *Kevin Murphy*, *Sebastian Thrun*, *Dieter Fox* and *Frank Dellaert* and their colleagues demonstrated the method in AI and robotics.
- *Tom Griffiths* and colleagues have studied SMC methods in cognitive psychology.

The 20th century - Tracking

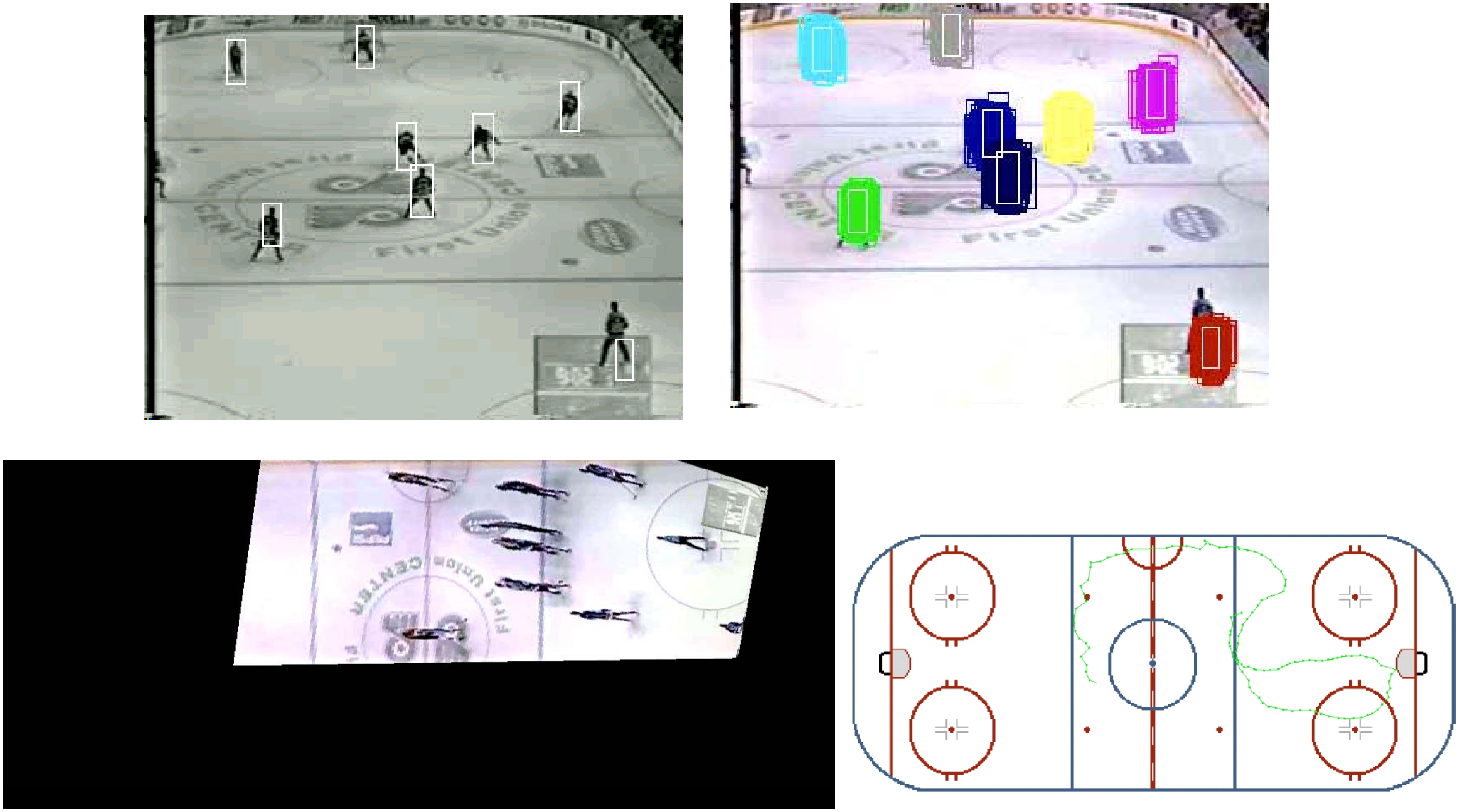
[Michael Isard & Andrew Blake (1996)]





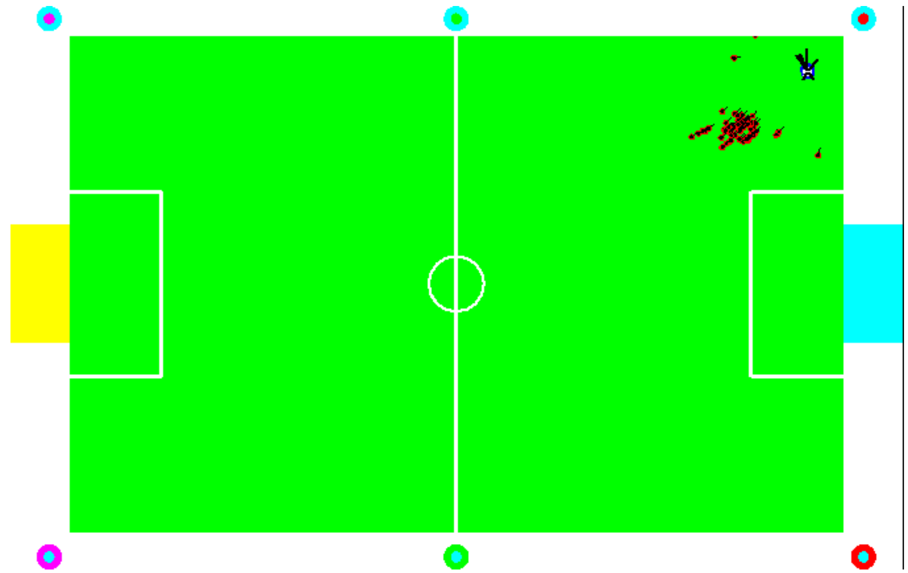


The 20th century - Tracking



[Boosted particle filter of Kenji Okuma, Jim Little & David Lowe]

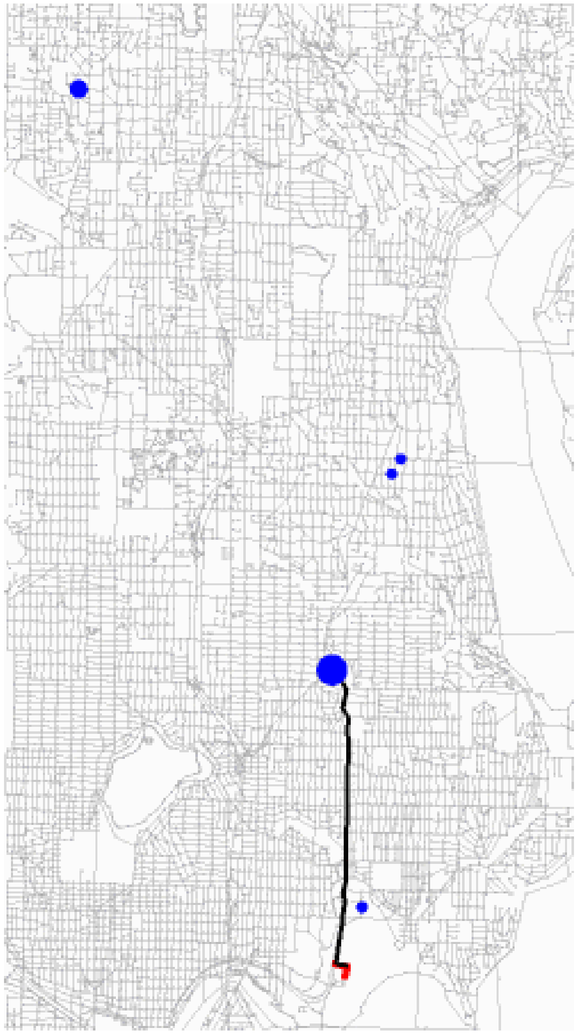
The 20th century – State estimation



[Dieter Fox]

http://www.cs.washington.edu/ai/Mobile_Robotics/mcl/

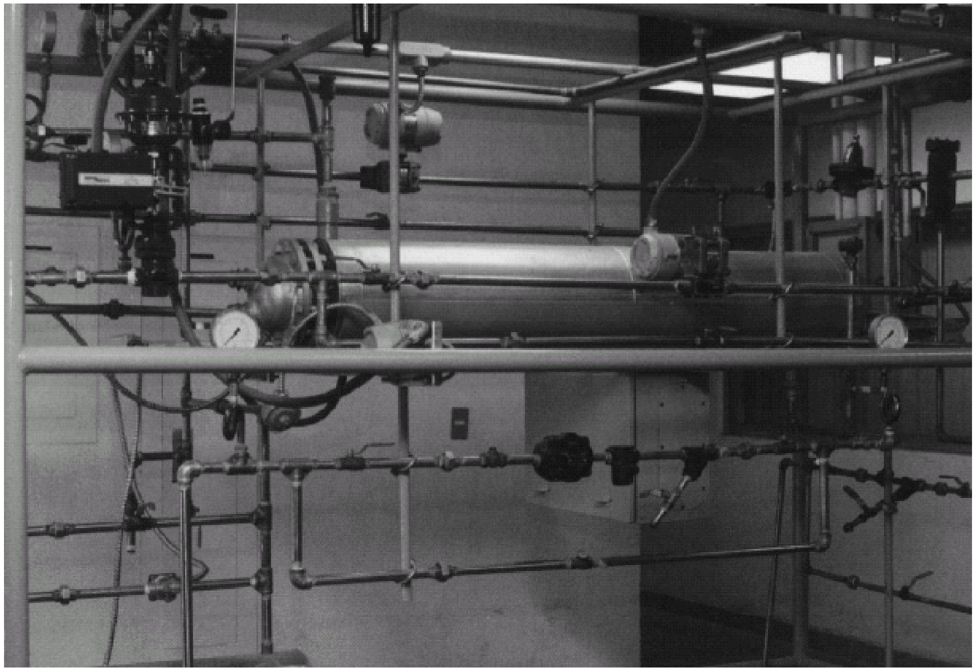
The 20th century – State estimation

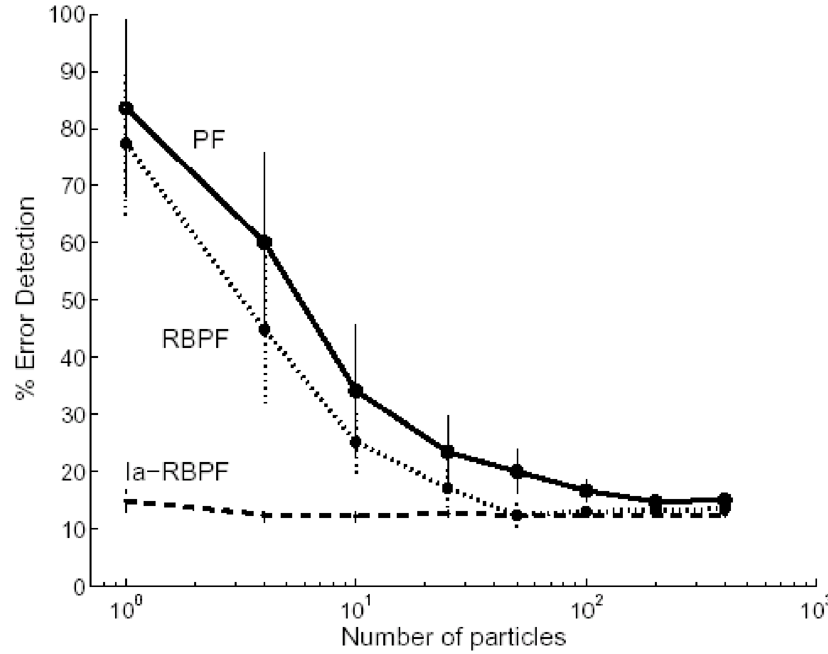


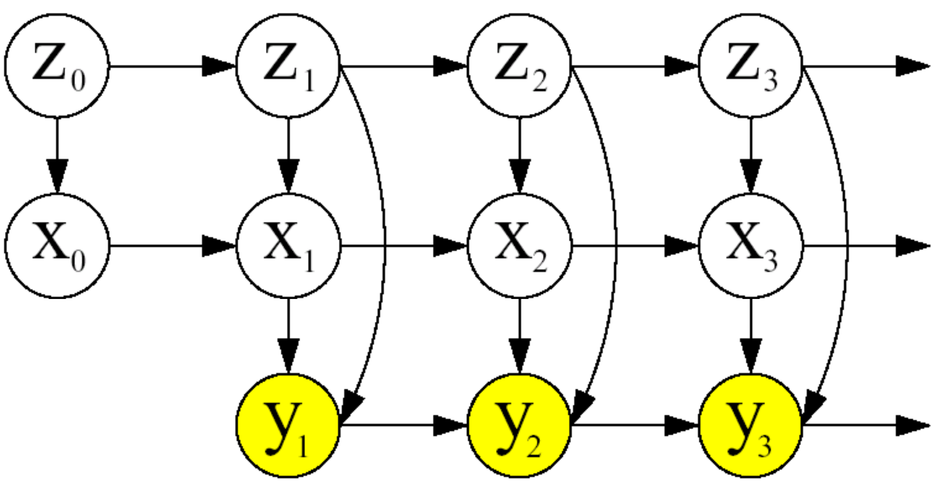
[Dieter Fox]

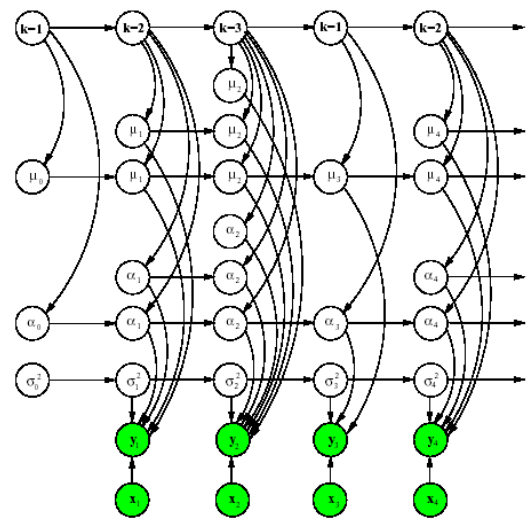
http://www.cs.washington.edu/ai/Mobile_Robotics/mcl/

The 20th century – State estimation

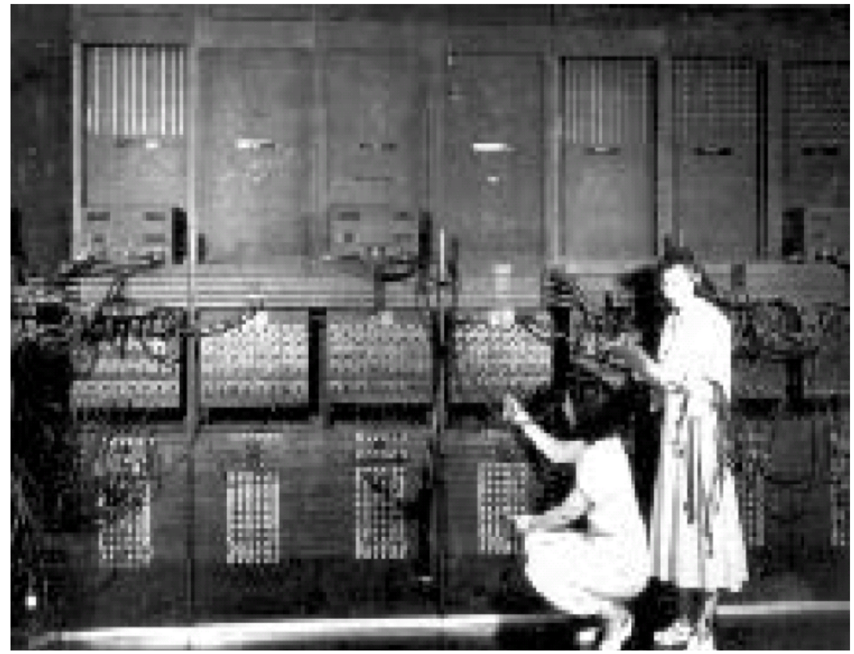


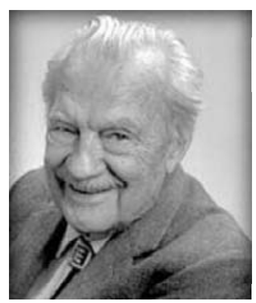






The 20th century – The birth





[Metropolis and Ulam, 1949]

Tutorial overview

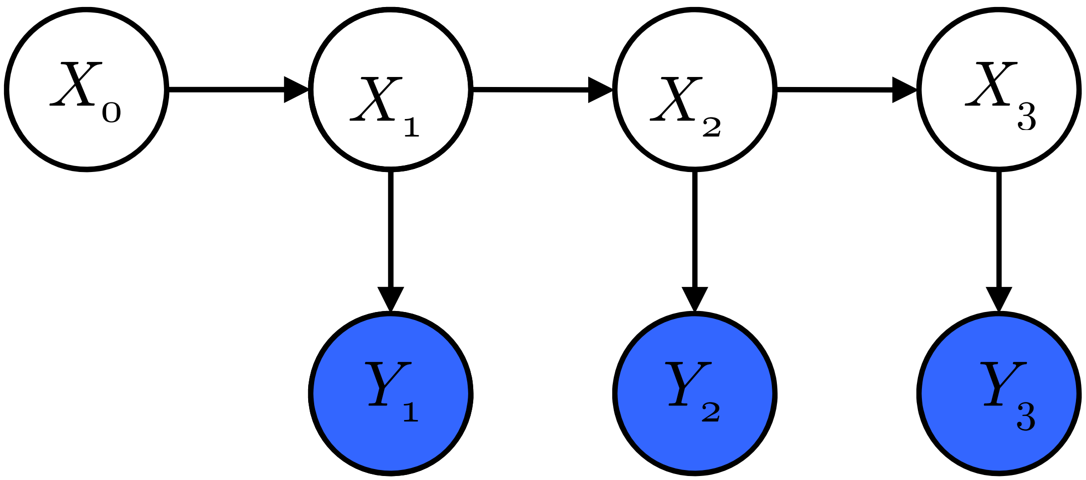
- **Introduction Nando – 10min**
- **Part I Arnaud – 50min**
  - Monte Carlo
  - Sequential Monte Carlo
  - Theoretical convergence
  - Improved particle filters
  - Online Bayesian parameter estimation
  - Particle MCMC
  - Smoothing
  - Gradient based online parameter estimation
- **Break 15min**
- **Part II NdF – 45 min**
  - Beyond state space models
  - Eigenvalue problems
  - Diffusion, protein folding & stochastic control
  - Time-varying Pitman-Yor Processes
  - SMC for static distributions
  - Boltzmann distributions & ABC

Arnaud’s slides will go here

Tutorial overview

- **Introduction Nando – 10min**
- **Part I Arnaud – 50min**
  - Monte Carlo
  - Sequential Monte Carlo
  - Theoretical convergence
  - Improved particle filters
  - Online Bayesian parameter estimation
  - Particle MCMC
  - Smoothing
  - Gradient based online parameter estimation
- **Break 15min**
- **Part II NdF – 45 min**
  - Beyond state space models
  - Eigenvalue problems
  - Diffusion, protein folding & stochastic control
  - Time-varying Pitman-Yor Processes
  - SMC for static distributions
  - Boltzmann distributions & ABC

Sequential Monte Carlo (recap)



$$P(X_0)P(X_1|X_0)P(Y_1|X_1)P(X_2|X_1)P(Y_2|X_2)P(X_3|X_2)P(Y_3|X_3) \propto P(X_{0:3}|Y_{1:3})$$

Sequences of distributions

- SMC methods can be used to sample approximately from any sequence of growing distributions $\{\pi_n\}_{n\geq 1}$

$$\pi_n(x_{1:n}) = \frac{f_n(x_{1:n})}{Z_n}$$

  where

  - $f_n : \mathcal{X}^n \to \mathbb{R}^+$ is known point-wise.
  - $Z_n = \int f_n(x_{1:n}) dx_{1:n}$

- We introduce a proposal distribution $q_n(x_{1:n})$ to approzimate Z_n:

$$Z_n = \int \frac{f_n(x_{1:n})}{q_n(x_{1:n})} q_n(x_{1:n}) dx_{1:n} = \int W_n(x_{1:n})\, q_n(x_{1:n}) dx_{1:n}$$

Importance weights

- Let us construct the proposal sequentially: Introduce $q_n\left(x_n \mid x_{1:n-1}\right)$ to sample component X_n given $X_{1:n-1} = x_{1:n-1}$.

$$q_n(x_{1:n}) = q(x_n|x_{1:n-1})\, q(x_{1:n-1})$$

- Then the importance weight becomes:

$$W_n = W_{n-1} \frac{f_n\left(x_{1:n}\right)}{f_{n-1}\left(x_{1:n-1}\right) q_n\left(x_n \mid x_{1:n-1}\right)}$$

$$W_n = \frac{f_n(x_{1:n})}{q_n(x_{1:n})} \frac{W_{n-1}}{W_{n-1}}$$

$$= \frac{f_n(x_{1:n})}{f_{n-1}(x_{1:n-1})} \frac{q_{n-1}(x_{1:n-1})}{q_n(x_{1:n})} W_{n-1}$$

$$=$$

SMC algorithm

1. Initialize at time $n = 1$

2. At time $n \geq 2$

- Sample $\overline{X}_n^{(i)} \sim q_n\left(x_n \middle| X_{1:n-1}^{(i)}\right)$ and augment $\overline{X}_{1:n}^{(i)} = \left(X_{1:n-1}^{(i)}, \overline{X}_n^{(i)}\right)$

- Compute the sequential weight

$$W_n^{(i)} \propto \frac{f_n\left(\overline{X}_{1:n}^{(i)}\right)}{f_{n-1}\left(\overline{X}_{1:n-1}^{(i)}\right) q_n\left(\overline{X}_n^{(i)} \middle| \overline{X}_{1:n-1}^{(i)}\right)}.$$

  Then the target approximation is:

$$\widetilde{\pi}_n\left(x_{1:n}\right) = \sum_{i=1}^{N} W_n^{(i)} \delta_{\overline{X}_{1:n}^{(i)}}\left(x_{1:n}\right)$$

- Resample $X_{1:n}^{(i)} \sim \widetilde{\pi}_n\left(x_{1:n}\right)$ to obtain $\widehat{\pi}_n\left(x_{1:n}\right) = \frac{1}{N}\sum_{i=1}^{N} \delta_{X_{1:n}^{(i)}}\left(x_{1:n}\right)$.

Example 1: Bayesian filtering

$$f_n(x_{1:n}) = p(x_{1:n}, y_{1:n}),\ \pi_n(x_{1:n}) = p(x_{1:n} | y_{1:n}),\ Z_n = p(y_{1:n}),$$

$$q_n(x_n | x_{1:n-1}) = f(x_n | x_{1:n-1}).$$

$$f_n(x_{1:n}) = P(x_{1:n}, y_{1:n}) = \prod_{k=1}^{n} P(x_k | x_{k-1}) P(y_k | x_k)$$

$$P(x_{1:n} | y_{1:n}) = \frac{P(x_{1:n}, y_{1:n})}{P(y_{1:n})}$$

$$w_n = w_{n-1} \frac{f_n(x_{1:n})}{f_{n-1}(x_{1:n-1})} \frac{1}{P(x_n | x_{n-1})}$$

$$= w_{n-1} P(y_n | x_n)$$

Example 2: Eigen-particles

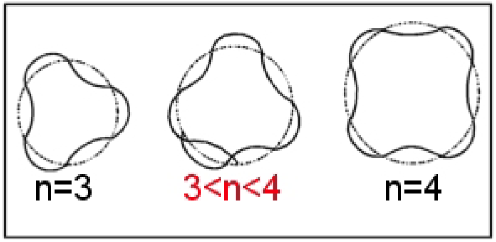
Computing eigen-pairs of exponentially large matrices and operators is an important problem in science. I will give two motivating examples:

i. Diffusion equation & Schrodinger's equation in quantum physics

ii. Transfer matrices for estimating the partition function of Boltzmann machines

Both problems are of enormous importance in physics and learning.

Quantum Monte Carlo



$$\left(-\frac{1}{2}\sum_{i=1}^{N}\nabla_i^2 + \sum_{i=1}^{N} v(\mathbf{r}_i) + \frac{1}{2}\sum_{i=1}^{N}\sum_{\substack{j=1\\(j\neq i)}}^{N} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} \right) \Psi(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N) = E\Psi(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N)$$

We can map this multivariable differential equation to an eigenvalue problem:

$$\int \psi(\mathbf{r}) K(\mathbf{s}|\mathbf{r}) d\mathbf{r} = \lambda \psi(\mathbf{s})$$

In the discrete case, this is the largest eigenpair of the $M \times M$ matrix A:

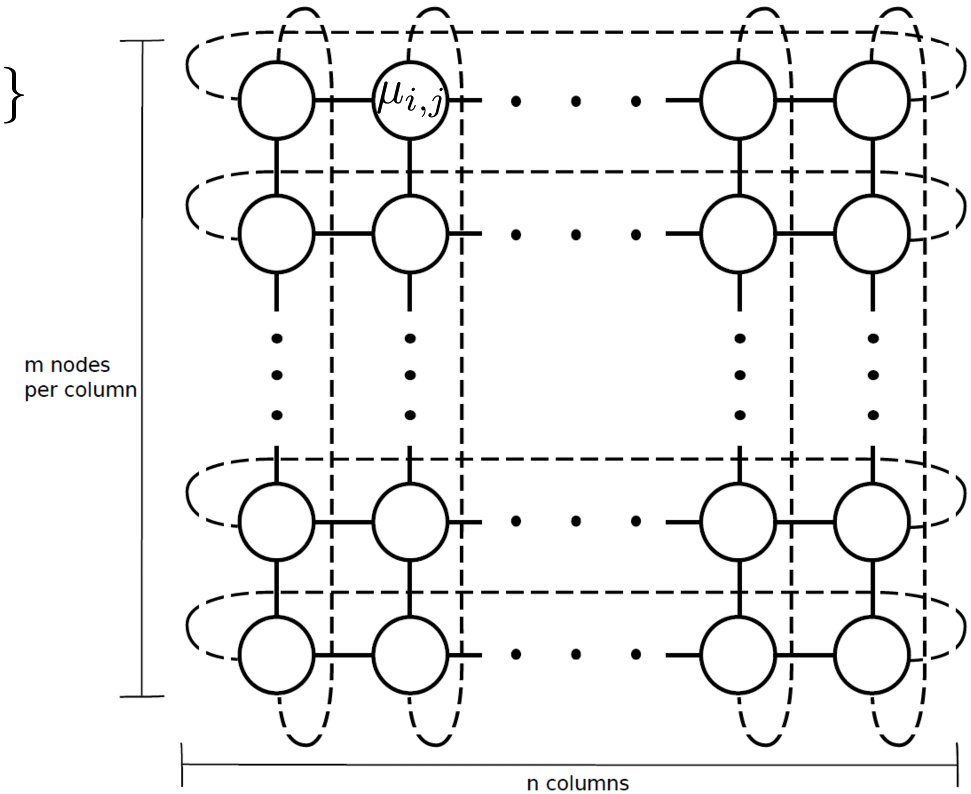
$$A\mathbf{x} = \lambda \mathbf{x} \quad \equiv \quad \sum_{i=1}^{M} x(r) a(r,s) = \lambda x(s) \ , \quad s = 1, 2, \ldots, M$$

where $a(r,s)$ is the entry of A at row r and column s.

[JB Anderson, 1975, I Kosztin et al, 1997]

Transfer matrices of Boltzmann Machines

$\mu_{i,j} \in \{-1, 1\}$



$$Z = \sum_{\{\boldsymbol{\mu}\}} \prod_{j=1}^{n} \exp\left(\nu \sum_{i=1}^{m} \mu_{i,j}\mu_{i+1,j} + \nu \sum_{i=1}^{m} \mu_{i,j}\mu_{i,j+1} \right)$$

$$= \sum_{\{\boldsymbol{\sigma}_1,\ldots,\boldsymbol{\sigma}_n\}} \prod_{j=1}^{n} A(\boldsymbol{\sigma}_j, \boldsymbol{\sigma}_{j+1}) = \sum_{k=1}^{2^m} \lambda_k^n$$

$$\boldsymbol{\sigma}_j = (\mu_{1,j}, \ldots, \mu_{m,j})$$

[see *e.g.* Onsager, Nimalan Mahendran]

Power method

Let A have M linearly independent eigenvectors, then any vector $\mathbf{v}$ may be represented as a linear combination of the eigenvectors of A: $\mathbf{v} = \sum_i c_i \mathbf{x}_i$, where c is a constant. Consequently, for sufficiently large n,

$$A^n\, \mathbf{v} \approx c_1 \lambda_1^n\, \mathbf{x}_1$$

$$Av = \sum_i c_i A x_i = \sum_i c_i \lambda_i x_i$$

$$A^n v = \sum_i c_i \lambda_i^n x_i$$

Particle power method

Succesive matrix-vector multiplication maps to Kernel-function multiplication (a path integral) in the continuous case:

$A^n v \quad f_n(x_n) = Z_n \pi_n(x_n$

$$\int \cdots \int v(\mathbf{x}_1) \prod_{k=2}^{n} K(\mathbf{x}_k|\mathbf{x}_{k-1}) d\mathbf{x}_{1:n-1} \approx c_1 \lambda_1^n \psi(\mathbf{x}_n)$$

The particle method is obtained by defining

$$f(\mathbf{x}_{1:n}) = v(\mathbf{x}_1) \prod_{k=2}^{n} K(\mathbf{x}_k|\mathbf{x}_{k-1})$$

Consequently $c\lambda_1^n \longrightarrow Z_n$ and $\psi(\mathbf{x}_n) \longrightarrow \pi(\mathbf{x}_n)$. The largest eigenvalue λ_1 of K is given by the ratio of successive partition functions:

$$\lambda_1 = \frac{Z_n}{Z_{n-1}}$$

The importance weights are

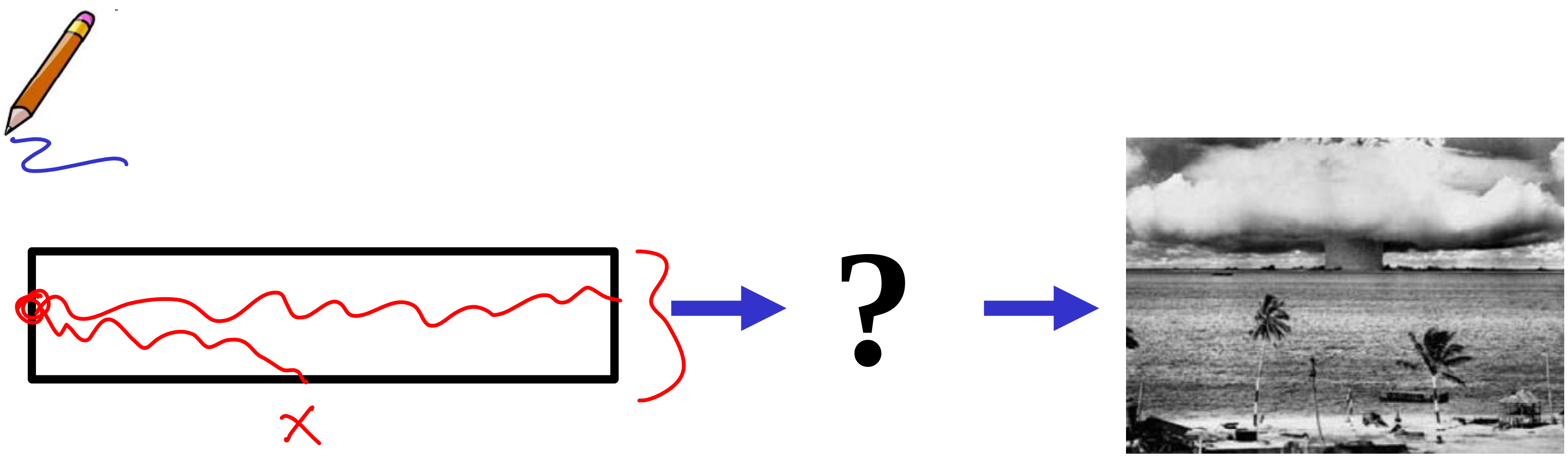
$$W_n = W_{n-1} \frac{v(\mathbf{x}_1) \prod_{k=2}^{n} K(\mathbf{x}_k|\mathbf{x}_{k-1})}{Q(\mathbf{x}_n|\mathbf{x}_{1:n}) v(\mathbf{x}_1) \prod_{k=2}^{n-1} K(\mathbf{x}_k|\mathbf{x}_{k-1})} = W_{n-1} \frac{K(\mathbf{x}_n|\mathbf{x}_{n-1})}{Q(\mathbf{x}_n|\mathbf{x}_{1:n})}$$

Example 3: Particle diffusion

- A particle $\{X_n\}_{n\geq 1}$ evolves in a random medium

$$X_1 \sim \mu(\cdot), \;\; X_{n+1} | X_n = x \sim p(\cdot | x).$$

- At time n, the probability of it being killed is $1 - g(X_n)$ with $0 \leq g(x) \leq 1$.
- One wants to approximate $\Pr(T > n)$.



Example 3: Particle diffusion

- Again, we obtain our familiar path integral:

$$\begin{aligned}\Pr(T > n) &= \mathbb{E}_{\mu}\left[\text{Probability of not being killed at } n \text{ given } X_{1:n}\right] \\ &= \int \cdots \int \mu(x_1) \prod_{k=2}^{n} p(x_k | x_{k-1}) \underbrace{\prod_{k=1}^{n} g(x_k)}_{\text{Probability to survive at } n} dx_{1:n}\end{aligned}$$

- Consider

$$\begin{aligned}f_n(x_{1:n}) &= \mu(x_1) \prod_{k=2}^{n} p(x_k | x_{k-1}) \prod_{k=1}^{n} g(x_k) \\ \pi_n(x_{1:n}) &= \frac{f_n(x_{1:n})}{Z_n} \text{ where } Z_n = \Pr(T > n)\end{aligned}$$

- SMC is then used to compute Z_n, the probability of not being killed at time n, and to approximate the distribution of the paths having survived at time n.

[Del Moral & AD, 2004]

Example 4: SAWs

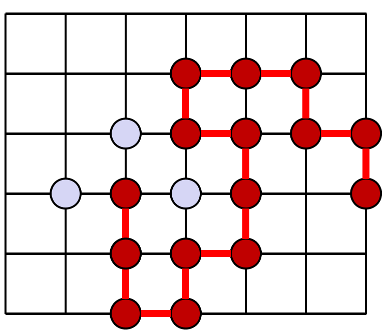
Goal: Compute the volume Z_n of a self-avoiding random walk, with uniform distribution on a lattice:

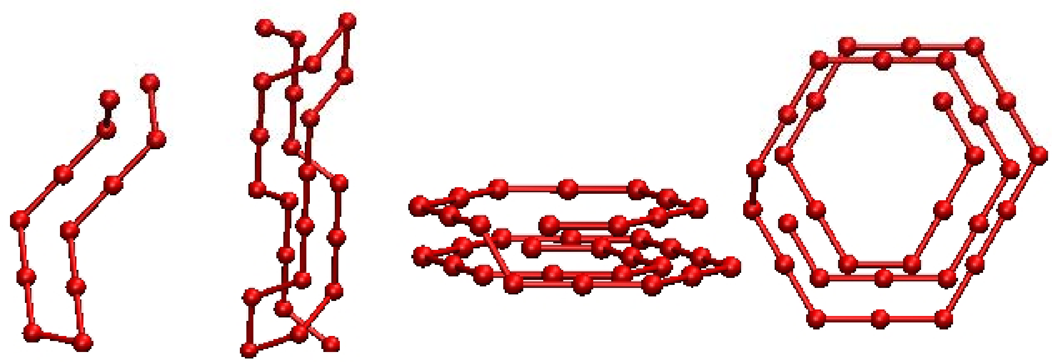
$$\pi_n(x_{1:n}) = Z_n^{-1} 1_{D_n}(x_{1:n})$$

where

$$\begin{aligned} D_n &= \{x_{1:n} \in E_n \text{ such that } x_k \sim x_{k+1} \text{ and } x_k \neq x_i \text{ for } k \neq i\}, \\ Z_n &= \text{cardinality of } D_n. \end{aligned}$$

SAWs on lattices are often used to study polymers and protein folding.





[See *e.g.* Peter Grassberger (PERM) & Alena Shmygelska; Rosenbluth Method]

Example 5: Stochastic control

- Consider a Fredholm equation of the 2nd kind (*e.g.* Bellman backup):

$$v(x_0) = r(x_0) + \int K(x_0, x_1) v(x_1) dx_1$$

- This expression can be easily transformed into a path integral (Von Neumann series representation):

$$v(x_0) = r(x_0) + \sum_{n=1}^{\infty} \int r(x_n) \prod_{k=1}^{n} K(x_{k-1}, x_k) dx_{1:n}$$

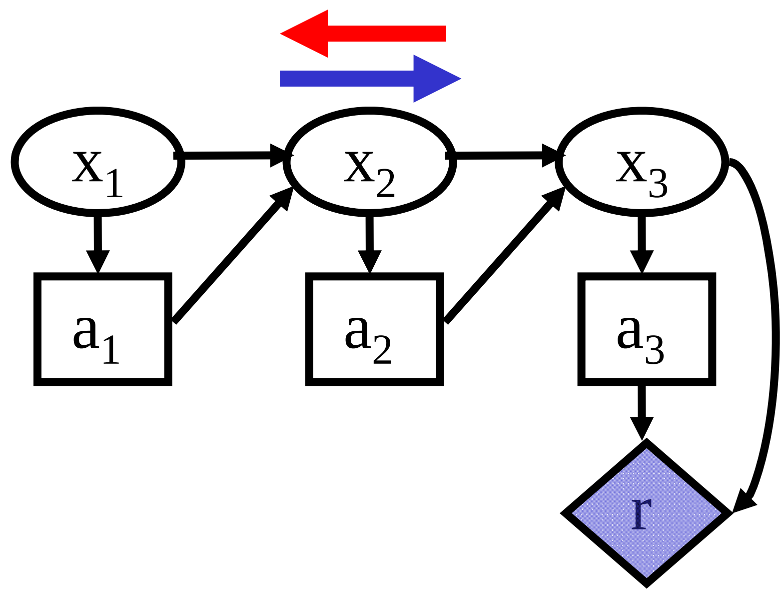
- The SMC sampler again follows by choosing

$$\begin{aligned} f_0(x_0) &= r(x_0) \\ f_n(x_{0:n}) &= r(x_n) \prod_{k=1}^{n} K(x_{k-1}, x_k) \end{aligned}$$

- In this case we have a trans-dimensional distribution, so we do a little bit more work when implementing the method.

[AD & Vladislav Tadic, 2005]

Particle smoothing can be used in the E step of the EM algorithm for MDPs



MDP posterior

$$\widetilde{p}_\theta\left(x_{0:k}, a_{0:k} \middle| k, r_k\right) = \frac{r\left(x_k, a_k\right) p_\theta\left(x_{0:k}, a_{0:k} \middle| k\right)}{\widetilde{p}_\theta\left(r_k \middle| k\right)}$$

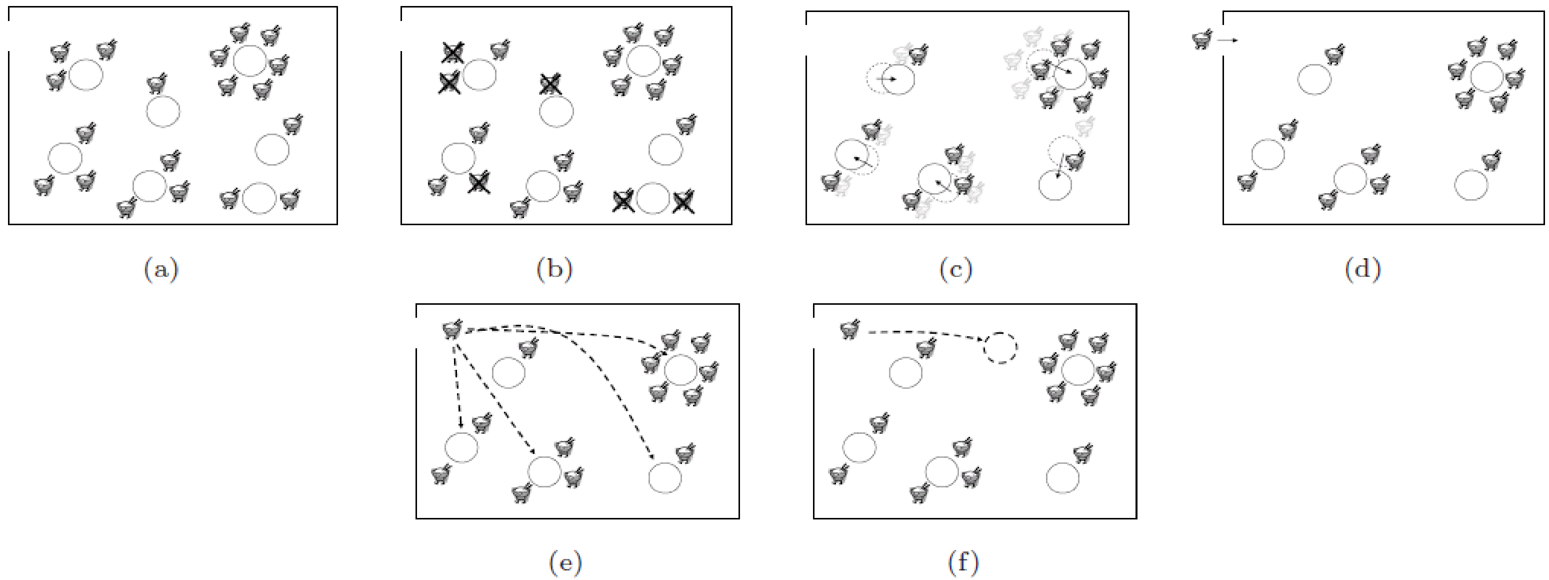
Likelihood: $r\left(x_k, a_k\right)$

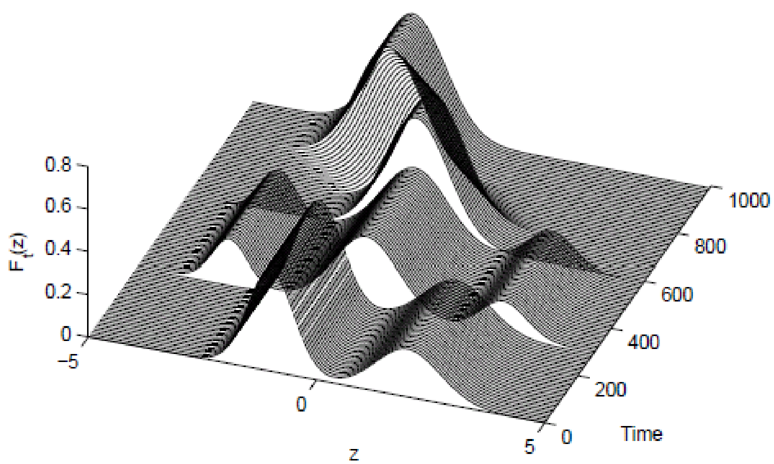
Prior: $p_\theta\left(x_{0:k}, a_{0:k} \middle| k\right)$

Marginal likelihood: $\widetilde{p}_\theta\left(r_k \middle| k\right)$

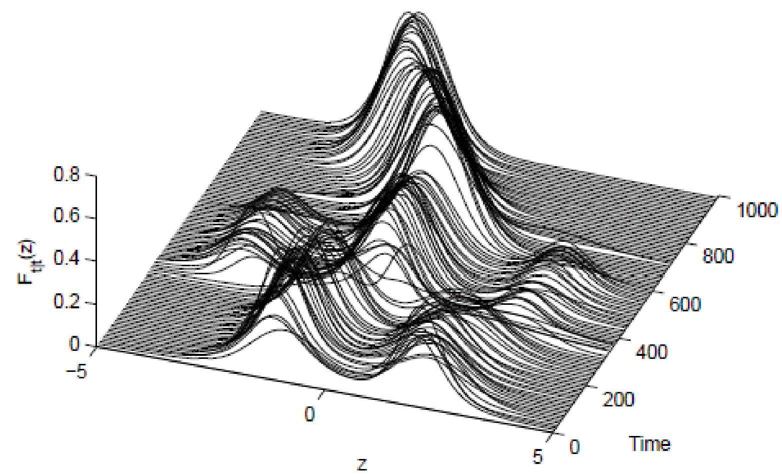
[See *e.g.* Matt Hoffman et al, 2007]

Example 6: Dynamic Dirichlet processes





(a) True density



(b) Estimated density

[Francois Caron, Manuel Davy & AD, 2007]

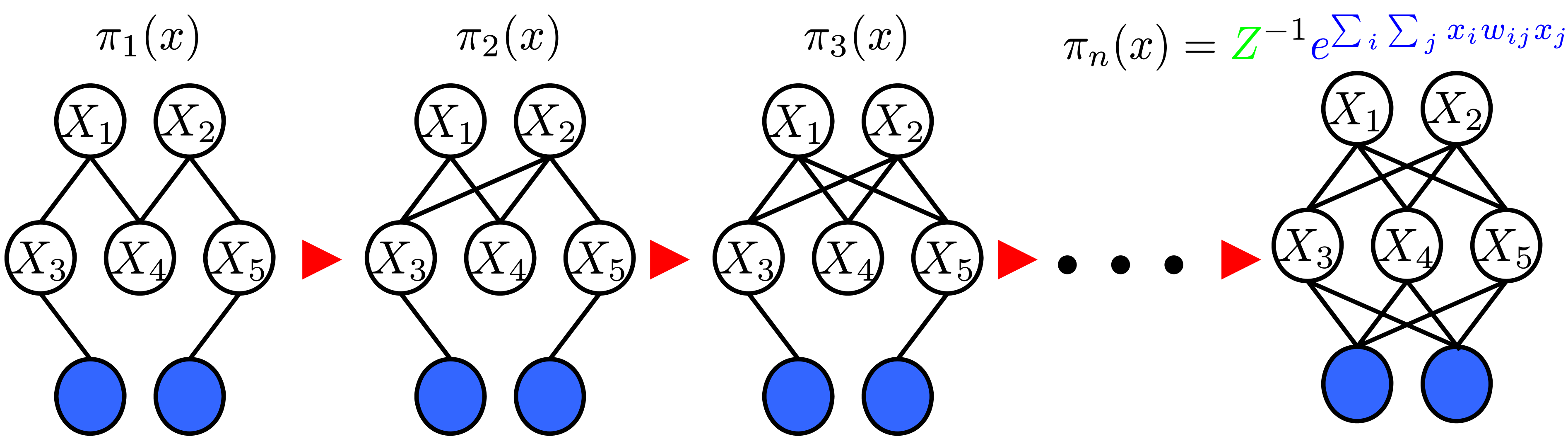
SMC for static models

- Let $\{\pi_n\}_{n\geq 1}$ be a sequence of probability distributions defined on $\mathcal{X}$ such that each $\pi_n(x)$ is known up to a normalizing constant, i.e.

$$\pi_n(x) = \underbrace{Z_n{}^{-1}}_{\text{unknown}} \underbrace{f_n(x)}_{\text{known}}$$

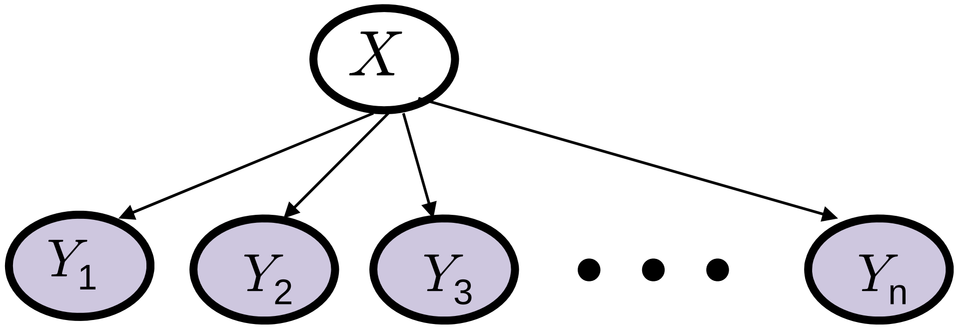
- We want to sample approximately from $\pi_n(x)$ and compute Z_n sequentially.

- This differs from the standard SMC, where $\pi_n(x_{1:n})$ is defined on $\mathcal{X}^n$.



Static SMC applications

- **Sequential Bayesian Inference:** $\pi_n(x) = p(x|y_{1:n})$.



- **Global optimization:** $\pi_n(x) \propto [\pi(x)]^{\eta_n}$ with $\{\eta_n\}$ increasing sequence such that $\eta_n \to \infty$.
- **Sampling from a fixed target** $\pi_n(x) \propto [\mu_1(x)]^{\eta_n} [\pi(x)]^{1-\eta_n}$ where μ_1 is easy to sample from. Use sequence $\eta_1 = 1 > \eta_{n-1} > \eta_n > \eta_{final} = 0$. Then $\pi_1(x) \propto \mu(x)$ and $\pi_{final}(x) \propto \pi(x)$
- **Rare event simulation** $\pi(A) \ll 1$: $\pi_n(x) \propto \pi(x) 1_{E_n}(x)$ with Z_1 known. Use sequence $E_1 = \mathcal{X} \supset E_{n-1} \supset E_n \supset E_{final} = A$. Then $Z_{final} = \pi(A)$.
- **Classical CS problems**: SAT, constraint satisfaction, computing volumes in high dimensions, matrix permanents and so on.

Static SMC derivation

- Construct an artificial distribution that is the product of the target distribution that we want to sample from and a backward kernel L:

$$\widetilde{\pi}_n(x_{1:n}) = Z_n^{-1} f_n(x_{1:n}), \text{ where } \quad f_n(x_{1:n}) = \underbrace{f_n(x_n)}_{\text{target}} \underbrace{\prod_{k=1}^{n-1} L_k(x_k | x_{k+1})}_{\text{artificial backward transitions}}$$

  such that $\pi_n(x_n) = \int \widetilde{\pi}_n(x_{1:n})\, dx_{1:n}$.

- The importance weights become:

$$\begin{aligned} W_n = \frac{f_n(x_{1:n})}{K_n(x_{1:n})} &= W_{n-1} \frac{K_{n-1}(x_{1:n-1})}{f_{n-1}(x_{1:n-1})} \frac{f_n(x_{1:n})}{K_n(x_{1:n})} \\ &= W_{n-1} \frac{f_n(x_n) L_{n-1}(x_{n-1}|x_n)}{f_{n-1}(x_{n-1}) K_n(x_n|x_{n-1})} \end{aligned}$$

- For the proposal $K(.)$, we can use any MCMC kernel.
- We only care about $\pi_n(x_n) = Z^{-1} f_n(x_n)$ so no degeneracy problem.

[Pierre Del Moral, AD, Ajay Jasra, 2006]

Static SMC algorithm

1. Initialize at time $n = 1$

2. At time $n \geq 2$

   (a) Sample $\overline{X}_n^{(i)} \sim K_n \left(x_n \middle| X_{n-1}^{(i)} \right)$ and augment $\overline{X}_{n-1:n}^{(i)} = \left(X_{n-1}^{(i)}, \overline{X}_n^{(i)} \right)$

   (b) Compute the importance weights

$$W_n^{(i)} = W_{n-1}^{(i)} \frac{f_n \left(\overline{X}_n^{(i)} \right) L_{n-1} \left(\overline{X}_{n-1}^{(i)} \middle| \overline{X}_n^{(i)} \right)}{f_{n-1} \left(\overline{X}_{n-1}^{(i)} \right) K_n \left(\overline{X}_n^{(i)} \middle| \overline{X}_{n-1}^{(i)} \right)}.$$

   Then the weighted approximation is

$$\widetilde{\pi}_n (x_n) = \sum_{i=1}^{N} W_n^{(i)} \delta_{\overline{X}_n^{(i)}} (x_n)$$

   (c) Resample $X_n^{(i)} \sim \widetilde{\pi}_n (x_n)$ to obtain $\widehat{\pi}_n (x_n) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_n^{(i)}} (x_n)$.

Static SMC: Choice of L

- A default (easiest) choice consists of using a π_n-invariant MCMC kernel K_n and the corresponding reversed kernel L_{n-1}:

$$L_{n-1}\left(x_{n-1}\middle|\,x_n\right)=\frac{\pi_n\left(x_{n-1}\right)K_n\left(x_n\middle|\,x_{n-1}\right)}{\pi_n\left(x_n\right)}$$

- In this case, the weights simplify to:

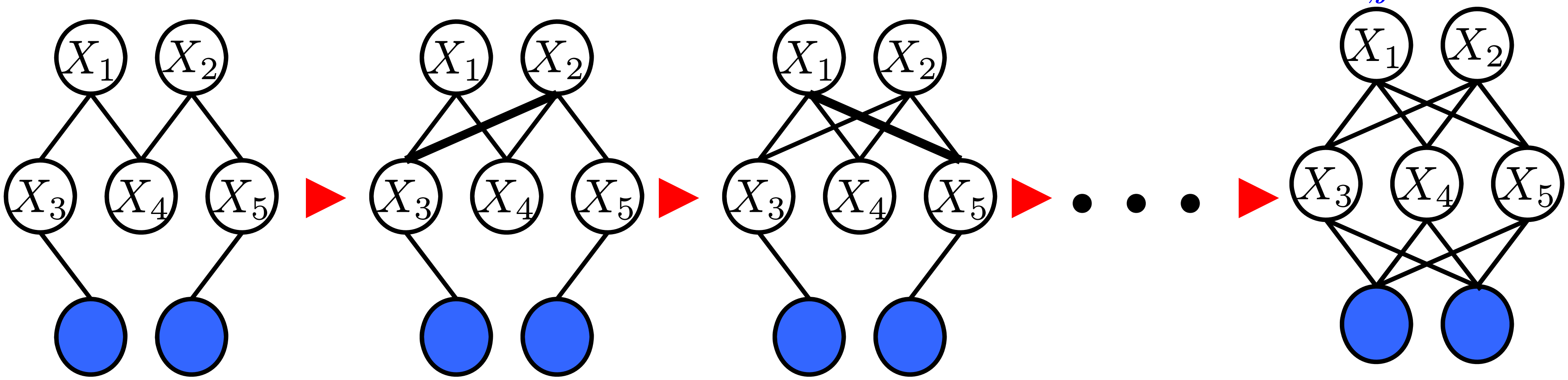
$$W_n^{(i)}=W_{n-1}^{(i)}\frac{f_n\left(X_{n-1}^{(i)}\right)}{f_{n-1}\left(X_{n-1}^{(i)}\right)}$$

- This particular choice appeared independently in physics and statistics (Jarzynski, 1997; Crooks, 1998; Gilks & Berzuini, 2001; Neal, 2001). In machine learning, it's often referred to as annealed importance sampling.

- Smarter choices of L can be sometimes implemented in practice.

Example 1: Deep Boltzmann machines

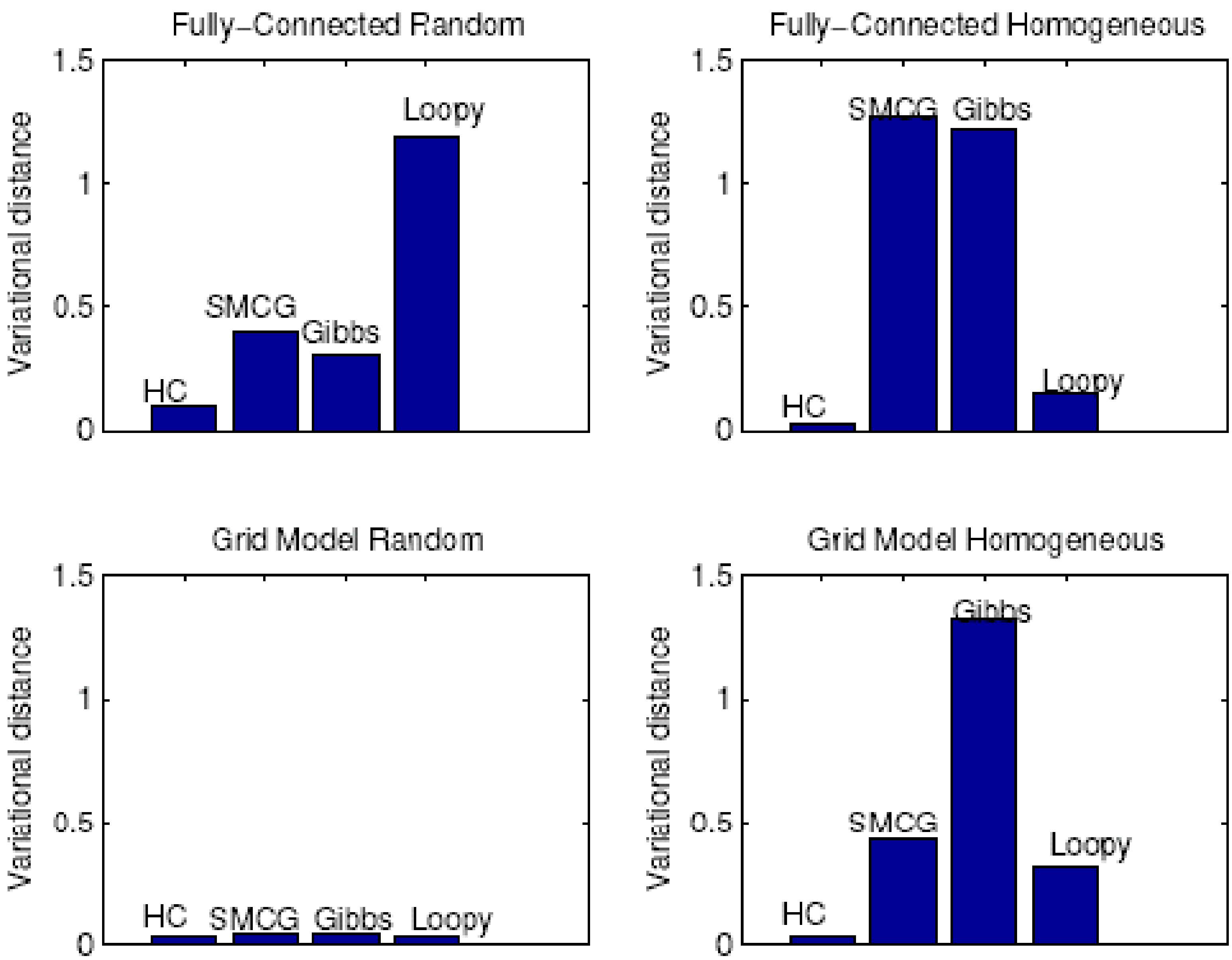
$\pi_1(x)$ $\pi_2(x)$ $\pi_3(x)$ $\pi_n(x) = Z^{-1} \prod_{i,j} \phi(x_i, x_j)$



$W_2^{(i)} \propto \phi(X_{2,3}^{(i)}, X_{2,2}^{(i)})$ $W_3^{(i)} \propto \phi(X_{3,1}^{(i)}, X_{3,5}^{(i)})$

[Firas Hamze, Hot coupling, 2005] [Peter Carbonetto, 2007, 2009]

Some results for undirected graphs



Example 2: ABC

- Consider a Bayesian model with prior $p(\theta)$ and likelihood $L(y|\theta)$ for data y. The likelihood is assumed to be intractable but we can sample from it.

- **ABC algorithm**:

  1. Sample $\theta^{(i)} \sim p(\theta)$
  2. Hallucinate data $Z^{(i)} \sim L\left(z|\theta^{(i)}\right)$
  3. Accept samples if hallucinations look like the data — if $d\left(y, Z^{(i)}\right) \leq \varepsilon$, where $d : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^{+}$ is a metric.

- The samples are approximately distributed according to:

$$\pi_{\varepsilon}(\theta, x|y) \propto p(\theta) L(x|\theta) 1_{d(y,z)\leq\varepsilon}$$

  The hope is that $\pi_{\varepsilon}(\theta|y) \approx \pi(\theta|y)$ for very small ε.

- Inefficient for ε small !

[Beaumont, 2002]

SMC samplers for ABC

- Define a sequence of artificial targets $\{\pi_{\varepsilon_n}(\theta|\, y)\}_{n=1,\ldots,P}$ where

$$\varepsilon_1 = \infty \geq \varepsilon_2 \geq \cdots \geq \varepsilon_P = \varepsilon.$$

- We can use SMC to sample from $\{\pi_{\varepsilon_n}(\theta|\, y)\}_{n=1,\ldots,P}$ by adopting a Metropolis-Hastings proposal kernel $K_n((\theta_n, z_n)|\,(\theta_{n-1}, z_{n-1}))$, with importance weights

$$W_n^{(i)} = W_{n-1}^{(i)} \frac{1_{d\left(y, Z_{n-1}^{(i)}\right) \leq \varepsilon_n}}{1_{d\left(y, Z_{n-1}^{(i)}\right) \leq \varepsilon_{n-1}}}$$

- Smarter algorithms have been proposed, which for example, compute the parameters ε_n and of K_n adaptively.

[Pierre Del Moral, AD, Ajay Jasra, 2009]

Final remarks

• SMC is a general, easy and flexible strategy for sampling from any arbitrary sequence of targets and for computing their normalizing constants.

• SMC is benefiting from the advent of GPUs. Anthony Lee Oxford

• SMC remains limited to moderately high-dimensional problems.

Thank you!

Nando de Freitas & Arnaud Doucet

Naïve SMC for static models

- At time $n-1$, you have particles $X_{n-1}^{(i)} \sim \pi_{n-1}(x_{n-1})$.
- Move the particles according to a transition kernel

$$X_n^{(i)} \sim K_n\left(x_n | X_{n-1}^{(i)}\right)$$

  hence marginally

$$X_n^{(i)} \sim \mu_n(x_n) \text{ where } \mu_n(x_n) = \int \pi_{n-1}(x_{n-1}) K_n(x_n | x_{n-1}) \, dx_{n-1}.$$

- Our target is $\pi_n(x_n)$ so the importance weight is

$$W_n^{(i)} \propto \frac{\pi_n\left(X_n^{(i)}\right)}{\mu_n\left(X_n^{(i)}\right)}.$$

- **Problem**: $\mu_n(x_n)$ does not admit an analytical expression in general cases.

test

test