Sequential Monte Carlo Methods

General Framwork for SISR, Growing a Polymer Chain in 2D
Convergence of SIS Vs. SISR
SMC for Static Problems

Prof. Nicholas Zabaras
University of Notre Dame
Notre Dame, IN, USA

Email: nzabaras@gmail.com
URL: https://www.zabaras.com/

October 24, 2017

Contents

References

- C.P. Robert & G. Casella, Monte Carlo Statistical Methods, Chapter 11
- J.S. Liu, Monte Carlo Strategies in Scientific Computing, Chapter 3, Springer-Verlag, New York.
- A. Doucet, N. De Freitas & N. Gordon (eds), Sequential Monte Carlo in Practice, Springer-Verlag: 2001
- A. Doucet, N. De Freitas, N.J. Gordon, An introduction to Sequential Monte Carlo, in SMC in Practice, 2001
- D. Wilkison, Stochastic Modelling for Systems Biology, Second Edition, 2006
- E. Ionides, Inference for Nonlinear Dynamical Systems, PNAS, 2006
- J.S. Liu and R. Chen, Sequential Monte Carlo methods for dynamic systems, JASA, 1998
- A. Doucet, Sequential Monte Carlo Methods, Short Course at SAMSI
- A. Doucet, Sequential Monte Carlo Methods & Particle Filters Resources
- Pierre Del Moral, Feynman-Kac models and interacting particle systems (SMC resources)
- A. Doucet, Sequential Monte Carlo Methods, Video Lectures, 2007
- N. de Freitas and A. Doucet, Sequential MC Methods, N. de Freitas and A. Doucet, Video Lectures, 2010

References

- M.K. Pitt and N. Shephard, Filtering via Simulation: Auxiliary Particle Filter, JASA, 1999
- A. Doucet, S.J. Godsill and C. Andrieu, On Sequential Monte Carlo sampling methods for Bayesian filtering, Stat. Comp., 2000
- J. Carpenter, P. Clifford and P. Fearnhead, An Improved Particle Filter for Non-linear Problems, IEE 1999.
- A. Kong, J.S. Liu & W.H. Wong, Sequential Imputations and Bayesian Missing Data Problems, JASA, 1994
- O. Cappe, E. Moulines & T. Ryden, Inference in Hidden Markov Models, Springer-Verlag, 2005
- W Gilks and C. Berzuini, Following a moving target: MC inference for dynamic Bayesian Models, JRSS B, 2001
- G. Poyadjis, A. Doucet and S.S. Singh, Maximum Likelihood Parameter Estimation using Particle Methods, Joint Statistical Meeting, 2005
- N Gordon, D J Salmond, AFM Smith, Novel Approach to nonlinear non Gaussian Bayesian state estimation, IEE, 1993
- Particle Filters, S. Godsill, 2009 (Video Lectures)
- R. Chen and J.S. Liu, Predictive Updating Methods with Application to Bayesian Classification, JRSS B, 1996

References

- C. Andrieu and A. Doucet, Particle Filtering for Partially Observed Gaussian State-Space Models, JRSS B, 2002
- R Chen and J Liu, Mixture Kalman Filters, JRSSB, 2000
- A Doucet, S J Godsill, C Andrieu, On SMC sampling methods for Bayesian Filtering, Stat. Comp. 2000
- N. Kantas, A.D., S.S. Singh and J.M. Maciejowski, An overview of sequential Monte Carlo methods for parameter estimation in general state-space models, in Proceedings IFAC System Identification (SySid) Meeting, 2009
- C. Andrieu, A.Doucet & R. Holenstein, Particle Markov chain Monte Carlo methods, JRSS B, 2010
- C. Andrieu, N. De Freitas and A. Doucet, Sequential MCMC for Bayesian Model Selection, Proc. IEEE Workshop HOS, 1999
- P. Fearnhead, MCMC, sufficient statistics and particle filters, JCGS, 2002
- G. Storvik, Particle filters for state-space models with the presence of unknown static parameters, IEEE Trans. Signal Processing, 2002

References

- C. Andrieu, A. Doucet and V.B. Tadic, Online EM for parameter estimation in nonlinear-non Gaussian state-space models, Proc. IEEE CDC, 2005
- G. Poyadjis, A. Doucet and S.S. Singh, Particle Approximations of the Score and Observed Information Matrix in State-Space Models with Application to Parameter Estimation, *Biometrika*, 2011
- C. Caron, R. Gottardo and A. Doucet, On-line Changepoint Detection and Parameter Estimation for Genome Wide Transcript Analysis, Technical report 2008
- R. Martinez-Cantin, J. Castellanos and N. de Freitas. Analysis of Particle Methods for Simultaneous Robot Localization and Mapping and a New Algorithm: Marginal-SLAM. International Conference on Robotics and Automation
- C. Andrieu, A.D. & R. Holenstein, Particle Markov chain Monte Carlo methods (with discussion), JRSS B, 2010
- A Doucet, Sequential Monte Carlo Methods and Particle Filters, List of Papers, Codes, and Viedo lectures on SMC and particle filters
- Pierre Del Moral, Feynman-Kac models and interacting particle systems

References

- P. Del Moral, A. Doucet and A. Jasra, Sequential Monte Carlo samplers, JRSSB, 2006
- P. Del Moral, A. Doucet and A. Jasra, Sequential Monte Carlo for Bayesian Computation, Bayesian Statistics, 2006
- P. Del Moral, A. Doucet & S.S. Singh, Forward Smoothing using Sequential Monte Carlo, technical report, Cambridge University, 2009
- A. Doucet, Short Courses Lecture Notes (A, B, C)
- P. Del Moral, Feynman-Kac Formulae, Springer-Verlag, 2004
- Sequential MC Methods, M. Davy, 2007
- A Doucet, A Johansen, Particle Filtering and Smoothing: Fifteen years later, in Handbook of Nonlinear Filtering (edts D Crisan and B. Rozovsky), Oxford Univ. Press, 2011
- A. Johansen and A. Doucet, A Note on Auxiliary Particle Filters, Stat. Proba. Letters, 2008.
- A. Doucet et al., Efficient Block Sampling Strategies for Sequential Monte Carlo, (with M. Briers & S. Senecal), JCGS, 2006.
- C. Caron, R. Gottardo and A. Doucet, On-line Changepoint Detection and Parameter Estimation for Genome Wide Transcript Analysis, Stat Comput. 2011.

GENERAL SEQUENTIAL IMPORTANCE SAMPLING RESAMPLING FRAMEWORK

Sequential Importance Sampling: General Framework

- Consider a sequence of probability distributions $\{\pi_n\}, n=1,2..$ defined on a sequence of spaces $\{E_n\}, n=1,2..$ where $E_1=\mathcal{X}$ and $E_n=E_{n-1}\times\mathcal{X}$.

- The distributions $\{\pi_n\}, n=1,2..$ are known up to a normalizing constant:

$$\pi_n(\boldsymbol{x}_{1:n}) = \frac{\gamma_n(\boldsymbol{x}_{1:n})}{Z_n}$$

- We want to estimate the expectations of functions $f_n : E_n \to \mathbb{R}$

$$\mathbb{E}_{\pi_n}(\varphi_n) = \int \varphi_n(\boldsymbol{x}_{1:n})\pi_n(\boldsymbol{x}_{1:n})d\boldsymbol{x}_{1:n}$$

and/or the normalizing constants Z_n.

- One can use MCMC to sample from $\{\pi_n\}, n=1,2..$ This calculation will be slow and cannot compute

$$\{Z_n\}, n=1,2..$$

Sequential Importance Sampling: General Framework

- We want to do these calculations sequentially starting with π_1 and Z_1 at step (time 1), then proceeding to π_2 and Z_2, etc.
- Sequential Monte Carlo (SMC) provides the means to do so as an alternative algorithm to MCMC.
- The key idea is that if π_{n-1} does not differ a lots from π_n, we should be able to reuse our estimate of π_{n-1} to approximate π_n.
- The only requirement for this general framework to work is:

$$\pi_n(\boldsymbol{x}_{1:n-1}) > 0 \Rightarrow \pi_{n-1}(\boldsymbol{x}_{1:n-1}) > 0$$

Sequential Importance Sampling

- We want to design an importance sampling method to approximate

$$\{\pi_n\}_{n\geq 1} \text{ and } \{Z_n\}_{n\geq 1}$$

- Assume that `at time 1', we have approximations $\hat{\pi}_1(x_1)$, $\hat{Z}_1$ using an importance density $q_1(x_1)$.

$$X_1^{(i)} \sim q_1(x_1)$$

$$\hat{\pi}_1(x_1)dx_1 = \sum_{i=1}^{N} W_1^{(i)}\, \delta_{X_1^{(i)}}(dx_1),\ where\ W_1^{(i)} \propto w\left(X_1^{(i)}\right)$$

$$\hat{Z}_1 = \frac{1}{N}\sum_{i=1}^{N} w_1\left(X_1^{(i)}\right)\ with$$

$$w_1(x_1) = \frac{\gamma_1(x_1)}{q_1(x_1)}$$

- We then resample $\left\{X_1^{(i)}, W_1^{(i)}\right\}$ to obtain new particles also denoted as $\left\{X_1^{(i)}\right\}$

Sequential Importance Sampling

- At `time 2', we want to approximate $\pi_2(\boldsymbol{x}_{1:2}), Z_2$ using an importance density $q_2(\boldsymbol{x}_{1:2})$.

- We want to reuse the samples $\boldsymbol{X}_1^{(i)}$ and $q_1(x_1)$ in building the importance sampling approximation for $\pi_2(\boldsymbol{x}_{1:2}), Z_2$.

- Let us select $q_2(\boldsymbol{x}_{1:2}) = q_1(x_1) q_2(x_2 \mid x_1)$

- To obtain $\boldsymbol{X}_{1:2}^{(i)} \sim q_2(\boldsymbol{x}_{1:2})$, we need to sample as follows:

$$X_2^{(i)} \mid X_1^{(i)} \sim q_2\left(x_2 \mid X_1^{(i)}\right)$$

- The importance sampling weight for this step is then:

$$w_2(\boldsymbol{x}_{1:2}) = \frac{\gamma_2(\boldsymbol{x}_{1:2})}{q_2(\boldsymbol{x}_{1:2})} = \frac{\gamma_2(\boldsymbol{x}_{1:2})}{q_1(x_1) q_2(x_2 \mid x_1)} =$$

$$= \frac{\gamma_1(x_1)}{q_1(x_1)} \frac{\gamma_2(\boldsymbol{x}_{1:2})}{\gamma_1(x_1) q_2(x_2 \mid x_1)} = \underbrace{w_1(x_1)}_{\text{Weight from step 1}} \underbrace{\frac{\gamma_2(\boldsymbol{x}_{1:2})}{\gamma_1(x_1) q_2(x_2 \mid x_1)}}_{\text{Incremental weight}}$$

Sequential Importance Sampling

- The normalized weights for step 2 are then given as:

$$W_2^{(i)} \propto \frac{\gamma_2\left(\boldsymbol{X}_{1:2}^{(i)}\right)}{\gamma_1\left(X_1^{(i)}\right) q_2\left(X_2^{(i)} \mid X_1^{(i)}\right)} \text{ (with resampling at Step 1)}$$

- Generalizing to step n, we can write (whether this process is Markov or not is unimportant):

$$\begin{aligned} q\left(\boldsymbol{x}_{1:n}\right) &= q_{n-1}\left(\boldsymbol{x}_{1:n-1}\right) q_n\left(x_n \mid \boldsymbol{x}_{1:n-1}\right) \\ &= q_1\left(x_1\right) q_2\left(x_2 \mid x_1\right) \cdots q_n\left(x_n \mid x_1, \cdots, x_{n-1}\right) \end{aligned}$$

- Thus if

$$\boldsymbol{X}_{1:n-1}^{(i)} \sim q_{n-1}\left(\boldsymbol{x}_{1:n-1}\right)$$

we sample X_n from

$$X_n^{(i)} \mid \boldsymbol{X}_{1:n-1}^{(i)} \sim q_n\left(x_n \mid \boldsymbol{X}_{1:n-1}^{(i)}\right)$$

Sequential Importance Sampling

- The weights for step *n* are then given as:

$$w_n(\boldsymbol{X}_{1:n}) = \frac{\gamma_n(\boldsymbol{X}_{1:n}^{(i)})}{q_n(\boldsymbol{X}_{1:n}^{(i)})} = \frac{\gamma_{n-1}(\boldsymbol{X}_{1:n-1}^{(i)})}{\underbrace{q_{n-1}(\boldsymbol{X}_{1:n-1}^{(i)})}_{w_{n-1}(\boldsymbol{X}_{1:n-1}^{(i)})}} \frac{\gamma_n(\boldsymbol{X}_{1:n}^{(i)})}{\gamma_{n-1}(\boldsymbol{X}_{1:n-1}^{(i)}) q_n(x_n \mid \boldsymbol{X}_{1:n-1}^{(i)})} = w_{n-1}(\boldsymbol{X}_{1:n-1}^{(i)}) \frac{\gamma_n(\boldsymbol{X}_{1:n}^{(i)})}{\gamma_{n-1}(\boldsymbol{X}_{1:n-1}^{(i)}) q_n(x_n \mid \boldsymbol{X}_{1:n-1}^{(i)})}$$

- Similarly the normalized weights are as follows:

$$W_n(\boldsymbol{X}_{1:n}^{(i)}) \propto \frac{\gamma_n(\boldsymbol{X}_{1:n}^{(i)})}{\gamma_{n-1}(\boldsymbol{X}_{1:n-1}^{(i)}) q_n(x_n \mid \boldsymbol{X}_{1:n-1}^{(i)})} \quad \text{(with resampling at step n - 1)}$$

- Contrary to the Hidden Markov Model considered in an earlier lecture, we may need to store all the paths $\{\boldsymbol{X}_{1:n}^{(i)}\}$ even if our interest is to only compute $\pi_n(x_n)$

- The computational cost maybe more significant than in the HMM case dependent on the specific form of the weights above.

Sequential Importance Sampling Algorithm

- The sequential importance sampling method thus can be defined as the following recursive procedure:

- When *n=1*, sample $X_1^{(i)} \sim q_1(.)$ and set: $w_1\left(X_1^{(i)}\right) = \dfrac{\gamma_1\left(X_1^{(i)}\right)}{q_1\left(X_1^{(i)}\right)}$

  Resample to obtain new particles $X_1^{(i)}$

- At step n (n>1)
  - Sample $X_n^{(i)} \sim q_n(x_n \mid \boldsymbol{X}_{1:n-1}^{(i)}), i = 1, \ldots, N$
  - Update the weight as:

$$w_n\left(X_{1:n}^{(i)}\right) = \frac{\gamma_n\left(\boldsymbol{X}_{1:n}^{(i)}\right)}{\gamma_{n-1}\left(\boldsymbol{X}_{1:n-1}^{(i)}\right) q_n\left(X_n^{(i)} \mid \boldsymbol{X}_{1:n-1}^{(i)}\right)}$$

At any step *n,* we then have:

$$\boldsymbol{X}_{1:n}^{(i)} \sim q_n(\boldsymbol{x}_{1:n}) \; and \; w_n\left(\boldsymbol{X}_n^{(i)}\right) = \frac{\gamma_n\left(\boldsymbol{X}_{1:n}^{(i)}\right)}{q_n\left(\boldsymbol{X}_{1:n}^{(i)}\right)}$$

Resample to obtain new particle population $\left\{\boldsymbol{X}_{1:n}^{(i)}, \dfrac{1}{N}\right\}$

Selection of Importance Density

$$w_n\left(\boldsymbol{x}_{1:n}\right) = w_{n-1}\left(\boldsymbol{x}_{1:n-1}\right) \cdot \frac{\gamma_n\left(\boldsymbol{x}_{1:n}\right)}{\gamma_{n-1}\left(\boldsymbol{x}_{1:n-1}\right) q_n\left(x_n \mid \boldsymbol{x}_{1:n-1}\right)}$$

- A (locally) optimal choice for the importance density is as follows:

$$q_n\left(x_n \mid \boldsymbol{x}_{1:n-1}\right) = \gamma_n\left(x_n \mid \boldsymbol{x}_{1:n-1}\right)$$

- With this selection note that the recursive weights take the form:

$$w_n\left(\boldsymbol{x}_{1:n}\right) = w_{n-1}\left(\boldsymbol{x}_{1:n-1}\right) \frac{1}{\gamma_{n-1}\left(\boldsymbol{x}_{1:n-1}\right)} \frac{\gamma_n\left(\boldsymbol{x}_{1:n}\right)}{\gamma_n\left(x_n \mid \boldsymbol{x}_{1:n-1}\right)} = w_{n-1}\left(\boldsymbol{x}_{1:n-1}\right) \frac{\gamma_n\left(\boldsymbol{x}_{1:n-1}\right)}{\gamma_{n-1}\left(\boldsymbol{x}_{1:n-1}\right)}$$

where

$$\gamma_n\left(\boldsymbol{x}_{1:n-1}\right) = \int_{x_n} \gamma_n\left(\boldsymbol{x}_{n-1}, x_n\right) dx_n$$

- Then, the incremental weight becomes

$$w_n = \frac{\gamma_n\left(\boldsymbol{x}_{1:n}\right)}{\gamma_{n-1}\left(\boldsymbol{x}_{1:n-1}\right) q_n\left(x_n \mid \boldsymbol{x}_{1:n-1}\right)} = \frac{\gamma_n\left(\boldsymbol{x}_{1:n-1}\right)}{\gamma_{n-1}\left(\boldsymbol{x}_{1:n-1}\right)} \left(\textit{independent of current state } x_n\right)$$

Locally Optimal Importance Distribution

$$q_n^{opt}\left(x_n \mid \boldsymbol{x}_{1:n-1}\right) = \gamma_n\left(x_n \mid \boldsymbol{x}_{1:n-1}\right)$$

- This is the most popular choice of the importance density.

$$w_n\left(\boldsymbol{x}_{1:n}\right) = w_{n-1}\left(\boldsymbol{x}_{1:n-1}\right)\frac{\gamma_n\left(\boldsymbol{x}_{1:n}\right)}{\gamma_{n-1}\left(\boldsymbol{x}_{1:n-1}\right)q_n\left(x_n \mid \boldsymbol{x}_{1:n-1}\right)} =$$

$$= w_{n-1}\left(\boldsymbol{x}_{1:n-1}\right)\frac{\gamma_n\left(\boldsymbol{x}_{1:n-1}\right)}{\gamma_{n-1}\left(\boldsymbol{x}_{1:n-1}\right)}$$

- Since it is often impossible to sample from $\pi_n\left(x_n \mid \boldsymbol{x}_{1:n-1}\right)$ or computing $\gamma_n\left(\boldsymbol{x}_{1:n-1}\right) = \int \gamma_n\left(\boldsymbol{x}_{1:n}\right)dx_n$, we need to approximate them.

- Note: The above approximations make sense if

$$\gamma_n\left(\boldsymbol{x}_{1:n-1}\right) = \int \gamma_n\left(\boldsymbol{x}_{1:n}\right)dx_n \approx \gamma_{n-1}\left(\boldsymbol{x}_{1:n-1}\right)$$

Sequential Importance Sampling

$$w_n\left(\boldsymbol{x}_{1:n}\right)=w_{n-1}\left(\boldsymbol{x}_{1:n-1}\right)\frac{\gamma_n\left(\boldsymbol{x}_{1:n}\right)}{\gamma_{n-1}\left(\boldsymbol{x}_{1:n-1}\right)q_n\left(x_n\mid\boldsymbol{x}_{1:n-1}\right)}=\underbrace{w_{n-1}\left(\boldsymbol{x}_{1:n-1}\right)\frac{\gamma_n\left(\boldsymbol{x}_{1:n-1}\right)}{\gamma_{n-1}\left(\boldsymbol{x}_{1:n-1}\right)}}_{\text{indepedent of current state } x_n}$$

where:

$$\gamma_n\left(\boldsymbol{x}_{1:n-1}\right)=\int\gamma_n\left(\boldsymbol{x}_{1:n}\right)dx_n$$

1. If w_n is too small, we reject the sample before we build it completely and restart.

2. This process needs to be implemented carefully to avoid introducing bias.

3. If resampling took place at step n-1, keep in mind that the above update is simply:

$$w_n\left(\boldsymbol{x}_{1:n}\right)=\frac{\gamma_n\left(\boldsymbol{x}_{1:n-1}\right)}{\gamma_{n-1}\left(\boldsymbol{x}_{1:n-1}\right)}$$

Convergence of Sequential Monte Carlo

- Consider the basic framework of Importance Sampling. The target distribution is $\pi_n(\boldsymbol{x}_{1:n})$, the un-normalized target distribution is denoted by $\gamma_n(\boldsymbol{x}_{1:n})$. The proposal distribution is $q_n(\boldsymbol{x}_{1:n})$. Thus the importance weight is

$$w_n(\boldsymbol{x}_{1:n}) = \frac{\gamma_n(\boldsymbol{x}_{1:n})}{q_n(\boldsymbol{x}_{1:n})}$$

and the normalization constant

$$Z_n = \int w_n(\boldsymbol{x}_{1:n}) q_n(\boldsymbol{x}_{1:n}) d\boldsymbol{x}_{1:n}$$

The estimator for Z_n is

$$\hat{Z}_n = \frac{1}{N} \sum_{i=1}^{N} w_n\left(X_{1:n}^{(i)}\right)$$

- We will compare the variances of $\hat{Z}_n / Z_n$ estimated with sequential importance sampling (SIS) and sequential Monte Carlo methods.

Convergence of Sequential Importance Sampling

- The variance of this estimator is

$$Var\left(\hat{Z}_n\right) = Var\left(\frac{1}{N}\sum_{i=1}^{N} w_n(\boldsymbol{x}_{1:n})\right) = \frac{1}{N^2} N \times Var\left(w_n(\boldsymbol{x}_{1:n})\right)$$

$$= \frac{1}{N}\int \left(w_n(\boldsymbol{x}_{1:n})\right)^2 q_n(\boldsymbol{x}_{1:n}) d\boldsymbol{x}_{1:n} - \frac{1}{N}\left[\int w_n(\boldsymbol{x}_{1:n}) q_n(\boldsymbol{x}_{1:n}) d\boldsymbol{x}_{1:n}\right]^2$$

$$= \frac{1}{N}\int \frac{\left(\gamma_n(\boldsymbol{x}_{1:n})\right)^2}{q_n(\boldsymbol{x}_{1:n})} d\boldsymbol{x}_{1:n} - \frac{1}{N}\left[\int \gamma_n(\boldsymbol{x}_{1:n}) d\boldsymbol{x}_{1:n}\right]^2$$

$$= \frac{1}{N} Z_n^2 \int \frac{\left(\pi_n(\boldsymbol{x}_{1:n})\right)^2}{q_n(\boldsymbol{x}_{1:n})} d\boldsymbol{x}_{1:n} - \frac{1}{N} Z_n^2$$

since

$$\gamma_n\left(\boldsymbol{x}_{1:n}\right) = Z_n \pi_n\left(\boldsymbol{x}_{1:n}\right)$$

Convergence of Sequential Importance Sampling

- Thus the relative variance

$$\frac{Var\left(\hat{Z}_n\right)}{Z_n^2} = \frac{1}{N}\left(\int \frac{\left(\pi_n\left(\boldsymbol{x}_{1:n}\right)\right)^2}{q_n\left(\boldsymbol{x}_{1:n}\right)} d\boldsymbol{x}_{1:n} - 1\right)$$

- **Example**
  Consider the case where

$$\pi_n\left(\boldsymbol{x}_{1:n}\right) = \prod_{k=1}^{n} \pi_n\left(x_k\right) = \prod_{k=1}^{n} \mathcal{N}\left(x_k; 0, 1\right)$$

$$\gamma_n\left(\boldsymbol{x}_{1:n}\right) = \prod_{k=1}^{n} \exp\left(-\frac{x_k^2}{2}\right)$$

$$Z_n = \left(2\pi\right)^{n/2}$$

- We select a proposal distribution

$$q_n\left(\boldsymbol{x}_{1:n}\right) = \prod_{k=1}^{n} q_k\left(x_k\right) = \prod_{k=1}^{n} \mathcal{N}\left(x_k; 0, \sigma^2\right)$$

Convergence of Sequential Importance Sampling

- In this case, the relative variance

$$\frac{Var(\hat{Z}_n)}{Z_n^2} = \frac{1}{N}\left(\int \frac{\left(\pi_n(x_{1:n})\right)^2}{q_n(x_{1:n})} d\boldsymbol{x}_{1:n} - 1\right)$$

$$= \frac{1}{N}\left(\int \prod_{k=1}^{n} \int \frac{\sigma}{\sqrt{2\pi}} \frac{\exp(-x_k^2)}{\exp\left(-\frac{1}{2}\frac{x_k^2}{\sigma^2}\right)} d\boldsymbol{x}_{1:n} - 1\right)$$

$$= \frac{1}{N}\left(\frac{(\sigma^2)^{n/2}}{(2\pi)^{n/2}} \prod_{k=1}^{n} \int \exp\left(-x_k^2\left(1 - \frac{1}{2\sigma^2}\right)\right) d\boldsymbol{x}_{1:n} - 1\right)$$

$$= \frac{1}{N}\left(\frac{(\sigma^2)^{n/2}}{(2\pi)^{n/2}} \prod_{k=1}^{n} \left(\frac{\pi}{\left(1 - \frac{1}{2\sigma^2}\right)}\right)^{1/2} - 1\right)$$

$$= \frac{1}{N}\left(\left(\frac{\sigma^4}{2\sigma^2 - 1}\right)^{n/2} - 1\right)$$

Convergence of Sequential Importance Sampling

- For example, if we select $\sigma^2 = 1.2$, the number of time steps *n=1000*, and expect/want a relative variance

$$\frac{Var(\hat{Z}_n)}{Z_n^2} = 0.01$$

then using

$$\frac{Var(\hat{Z}_n)}{Z_n^2} = \frac{1}{N}\left(\left(\frac{\sigma^4}{2\sigma^2 - 1}\right)^{n/2} - 1\right)$$

the number of needed samples comes to be:

$$N \approx 1.31 \times 10^8$$

Convergence of Sequential Importance Sampling Resampling

- If multinomial resampling is used at every step, the associated SMC estimate of $\hat{Z}_n/Z_n$ has an asymptotic variance as (Doucet A and Johansen A, 2008, pp16)

$$Var\left(\frac{\hat{Z}_n}{Z_n}\right) = \frac{1}{N}\left[\int \frac{\pi_n^2(x_1)}{q_1(x_1)}dx_1 - 1 + \sum_{k=2}^{n}\left(\int \frac{\pi_n^2(\boldsymbol{x}_{1:k})}{\pi_{k-1}(\boldsymbol{x}_{1:k-1})q_k(x_k|\boldsymbol{x}_{1:k-1})}d\boldsymbol{x}_{k-1:k} - 1\right)\right]$$

- **Example** (continue)

$$\pi_n(\boldsymbol{x}_{1:n}) = \prod_{k=1}^{n}\pi_n(x_k) = \prod_{k=1}^{n}\mathcal{N}(x_k;0,1)$$

$$\gamma_n(\boldsymbol{x}_{1:n}) = \prod_{k=1}^{n}\exp\left(-\frac{x_k^2}{2}\right)$$

$$Z_n = (2\pi)^{n/2}$$

$$q_n(\boldsymbol{x}_{1:n}) = \prod_{k=1}^{n}q_k(x_k) = \prod_{k=1}^{n}\mathcal{N}(x_k;0,\sigma^2)$$

Convergence of Sequential Importance Sampling Resampling

- In this case,

$$\int \frac{\pi_n^2(\boldsymbol{x}_{1:k})}{\pi_{k-1}(\boldsymbol{x}_{1:k-1})q_k(x_k \mid \boldsymbol{x}_{1:k-1})} d\boldsymbol{x}_{k-1:k} = \int \frac{\pi_n^2(\boldsymbol{x}_{1:k-1})\pi_n^2(x_k)}{\pi_{k-1}(\boldsymbol{x}_{1:k-1})q_k(x_k \mid \boldsymbol{x}_{1:k-1})} d\boldsymbol{x}_{k-1:k} = \int \frac{\pi_n^2(x_k)}{q_k(x_k)} \pi_n(\boldsymbol{x}_{1:k-1}) d\boldsymbol{x}_{k-1:k}$$

$$\leq \int \frac{\pi_n^2(x_k)}{q_k(x_k)} dx_k, \text{since} \int \pi_n(x_{k-1}) dx_{k-1} = 1 \text{ and } \pi_n(\boldsymbol{x}_{1:k-2}) = \prod_{k=1}^{k-2} \mathcal{N}(x_k \mid 0,1) \leq 1$$

- Thus

$$Var\left(\frac{\hat{Z}_n}{Z_n}\right) \leq \frac{1}{N}\left[\sum_{k=1}^{n}\left(\int \frac{\pi_n^2(x_k)}{q_k(x_k)} dx_k - 1\right)\right]$$

- Substitute $\pi_n(x_k) = \mathcal{N}(x_k; 0,1)$ and $q_k(x_k) = \mathcal{N}(x_k; 0, \sigma^2)$ into above formula, we have

$$Var\left(\frac{\hat{Z}_n}{Z_n}\right) \leq \frac{n}{N}\left[\left(\frac{\sigma^4}{2\sigma^2 - 1}\right)^{1/2} - 1\right]$$

Convergence of Sequential Importance Sampling Resampling

- We apply SMC to previous example, and assume the same parameters : $\sigma^2 = 1.2$, n=1000, and .

$$\frac{V_{SMC}(\hat{Z}_n)}{Z_n^2} = 0.01$$

- In this case, we only need approximately N=1420 samples to achieve the same accuracy !

GROWING A POLYMER IN A 2D LATTICE: A SIS Application

- J. M. Hammersley and K. W. Morton, Poor Man's Monte Carlo, *Journal of the Royal Statistical Society. Series B (Methodological)* Vol. 16, No. 1 (1954), pp. 23-38

Example: Growing a Polymer in a 2D Lattice

- This example is a simulation of self-avoiding random walks where *E* is a finite lattice and

$$\gamma_n\left(\boldsymbol{x}_{1:n}\right)=\mathbb{I}_{A_n}\left(\boldsymbol{x}_{1:n}\right)$$

$$A=\left\{\boldsymbol{x}_{1:n}\in E^n:\forall i,j\in\{1,2,...,n\},x_i\neq x_j \;\; for\; all\; i\neq j\right\}$$

- In this case, $\pi_n\left(x_n\mid\boldsymbol{x}_{1:n-1}\right)$ is a uniform distribution over the free (available for move) neighbors.

- We will see an example next in the growth of a 2D polymer chain in a two-dimensional lattice.

Example: Growing a Polymer in a 2D Lattice

- Given the length N of the chain ($N+1$ molecules), we can assume a uniform probability density for the polymer chains, i.e. the target distribution of the positions $\boldsymbol{x}$ of all molecules is

$$\pi(\boldsymbol{x}) = \frac{1}{Z_N}$$

  where Z_N is the (unknown) number of all possible polymer chains of length N.

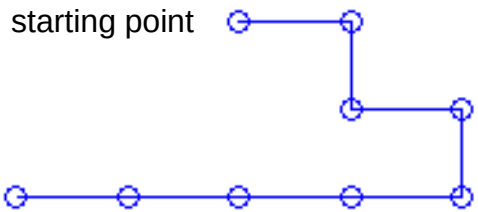
- **We want to sample different chains of length N** as we are interested to compute statistics such as the mean square extension of the chain.

- In the self-avoid random walk, we sample directly from the target distribution (uniform) and each valid polymer chain is generated with equal probability.

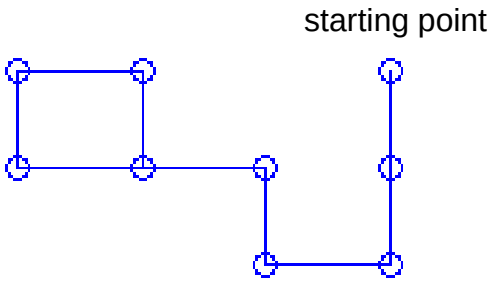
- In the SIS approach, we sample sequentially from importance sampling densities.

Example: Growing a Polymer in a 2D Lattice

- In our model, the realization of a chain polymer of length N is fully characterized by the positions of all of its molecules (starting from $x_0=(0,0)$), $\boldsymbol{x}=(x_1, x_2, ..., x_N)$, where x_i is a point in the 2-D lattice space, i.e. $x_i=(a,b)$ where a and b are integers.
- The distance between x_i and x_{i+1} has to be exactly *1*
- No molecules have the same positions (superposition is not permitted)



polymer chain



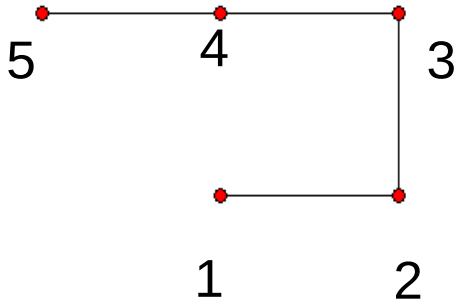
NOT a polymer chain

Example: Growing a Polymer in a 2D Lattice

- The most naive way of simulating the chain paths is to start a random walk at *(0,0).*
  - At each step *n*, we choose one of the three allowed neighboring positions to go with equal probability (we are not allowed to fall back).
  - If that position has already been taken up earlier, we start a new chain from *(0,0)* again. Otherwise, we keep going on until the presumed length *N* is reached.

Self-Avoid Random Walk

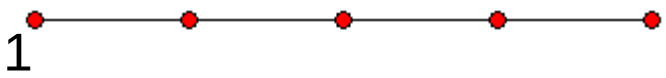
- Suppose we already have 4 points, and we are to generate the 5th point.



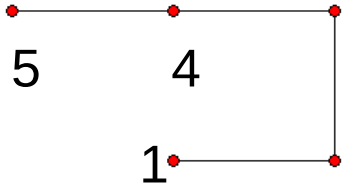
- We restrict that the chain cannot move backward. Then, there are three possible positions for point 5 (note that we don't consider whether the position is occupied by previous points).
- Then we uniformly select a position to place point 5 (with probability 1/3). If the position we randomly selected coincides with point 1, the generation fails, and we restart from the very beginning.
- A successful chain generated in this way is subject to uniform probability $1/Z_N$, where Z_N is all possible configurations of chains of length N.

Self-Avoid Random Walk

- Using this method, any two chains are generated with equal probability e.g.



probability: 1/4*1/3*1/3*1/3=1/108



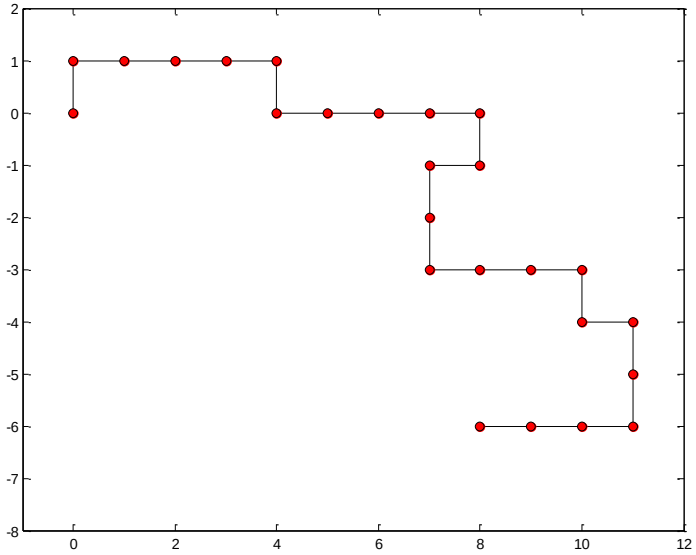
probability: 1/4*1/3*1/3*1/3=1/108

- At point 4, we select point 5 with probability 1/3. If point 5 lies in the same position with point 1, we just reject this sample.

- In another words, in computing the probabilities, we don't care whether any of the current neighboring positions is occupied or not.

Example: Growing a Polymer in a 2D Lattice

- One sample with *25* molecules and length *24* (starting from (*0,0*))



- This simulation procedure is apparently inefficient. Given the length *N*, the number of successes over the number of trials of obtaining a acceptable chain decreases exponentially as a function of *N*.
  - For *N=20*, this rate is approximately *21.6%;* and
  - For *N=48*, the acceptance rate is as low as *0.79%.*

Importance Sampling Approach

- Constructing the polymer chain in a recursive way can improve the efficiency of simulation.

- Starting at position $x_1=(0,0)$, the position of the second point is fixed with probability

$$q_2\left[x_2=(i_2,j_2)\,|\,x_1\right]=\frac{1}{\eta_1}\quad,\quad \eta_1=4$$

- The third point is fixed with probability

$$q_3\left[x_3=(i_3,j_3)\,|\,x_1,x_2\right]=\frac{1}{\eta_2}\quad,\quad \eta_2=3$$

- The N-th point is fixed with probability

$$q_N\left[x_N=(i_N,j_N)\,|\,x_1,x_2,\cdots,x_{N-1}\right]=\frac{1}{\eta_{N-1}}$$

where (i_N,j_N) is one of the unoccupied neighbors of x_{N-1} and η_{N-1} is the total number of such unoccupied neighbors.

Importance Sampling Approach

- Therefore, the importance distribution

$$q_N\left(\boldsymbol{x}_{1:N}\right)=q_1\left(x_1\right)q_2\left(x_2\mid x_1\right)q_3\left(x_3\mid \boldsymbol{x}_{1:2}\right)\cdots q_N\left(x_N\mid \boldsymbol{x}_{1:N-1}\right)$$

$$\propto \frac{1}{\eta_1\eta_2\cdots\eta_{N-1}}$$

- Because the target distribution $\pi\left(\boldsymbol{x}_{1:N}\right)=\frac{1}{Z_N}\propto\gamma\left(\boldsymbol{x}_{1:N}\right)=1$, the un-normalized importance weight

$$w\left(\boldsymbol{x}_{1:N}\right)=\frac{\gamma\left(\boldsymbol{x}_{1:N}\right)}{q\left(\boldsymbol{x}_{1:N}\right)}=\frac{1}{q_1\left(x_1\right)q_2\left(x_2\mid x_1\right)\cdots q_N\left(x_N\mid \boldsymbol{x}_{1:N-1}\right)}$$

- In a recursive form,

$$w_n\left(\boldsymbol{x}_{1:n}\right)=w_{n-1}\left(\boldsymbol{x}_{1:n-1}\right)\frac{1}{q_n\left(x_n\mid \boldsymbol{x}_{1:n-1}\right)}=w_{n-1}\left(\boldsymbol{x}_{1:n-1}\right)\eta_{n-1}$$

Importance Sampling Approach

- In the SIS, we select uniform distributions on the space of chains of length *n, n=1,..,N*.
- Since Z_n is unknown, we only know these distributions up to a constant.

$$\gamma_n(\boldsymbol{x}_{1:n}) = 1,\ \pi_n(\boldsymbol{x}_{1:n}) = 1/Z_n$$

- Let $\eta_n(\boldsymbol{x}_{1:n})\ (\eta_n \le 3)$ be the number of available nodes after step *n*.
- We can calculate the conditionals $\pi_n(x_n \mid \boldsymbol{x}_{1:n-1})$ as follows:

$$Since\ \pi_n(\boldsymbol{x}_{1:n-1}) = \sum_{x_n} \pi_n(\boldsymbol{x}_{1:n-1}, x_n) = \frac{\eta_{n-1}}{Z_n} \Rightarrow$$

$$\pi_n(x_n \mid \boldsymbol{x}_{1:n-1}) = \frac{\pi_n(\boldsymbol{x}_{1:n})}{\pi_n(\boldsymbol{x}_{1:n-1})} = \frac{1/Z_n}{\eta_{n-1}/Z_n} = \frac{1}{\eta_{n-1}}$$

Example: Growing a Polymer in a 2D Lattice

- We can now use this as our importance sampling density at step *t* (locally optimal approximation):

$$q_n\left(x_n \mid \boldsymbol{x}_{1:n-1}\right) = \gamma_n\left(x_n \mid \boldsymbol{x}_{1:n-1}\right) = \frac{1}{\eta_{n-1}}$$

i.e. we will select one of the available neighbors with equal probability.

- The importance sampling weights are now:

$$w_n\left(\boldsymbol{x}_{1:n}\right) = w_{n-1}\left(\boldsymbol{x}_{1:n-1}\right)\frac{\gamma_n\left(\boldsymbol{x}_{1:n-1}\right)}{\gamma_{n-1}\left(\boldsymbol{x}_{1:n-1}\right)} = w_{n-1}\left(\boldsymbol{x}_{1:n-1}\right)\eta_{n-1} \; (\textit{for } \eta_{n-1} > 0, \textit{otherwise} \;\; w_n = 0)$$

- Each sample $\boldsymbol{x}_{1:N}$ will finally have the following weight:

$$w_N\left(\boldsymbol{x}_{1:N}\right) = \eta_1\eta_2 \ldots \eta_{N-1}$$

Example: Growing a Polymer in a 2D Lattice

- We build the path sequentially. Suppose at step *n*, if all four neighbors have been visited before, the procedure is terminated; otherwise, we select one of the available positions (not visited before) with equal probability.

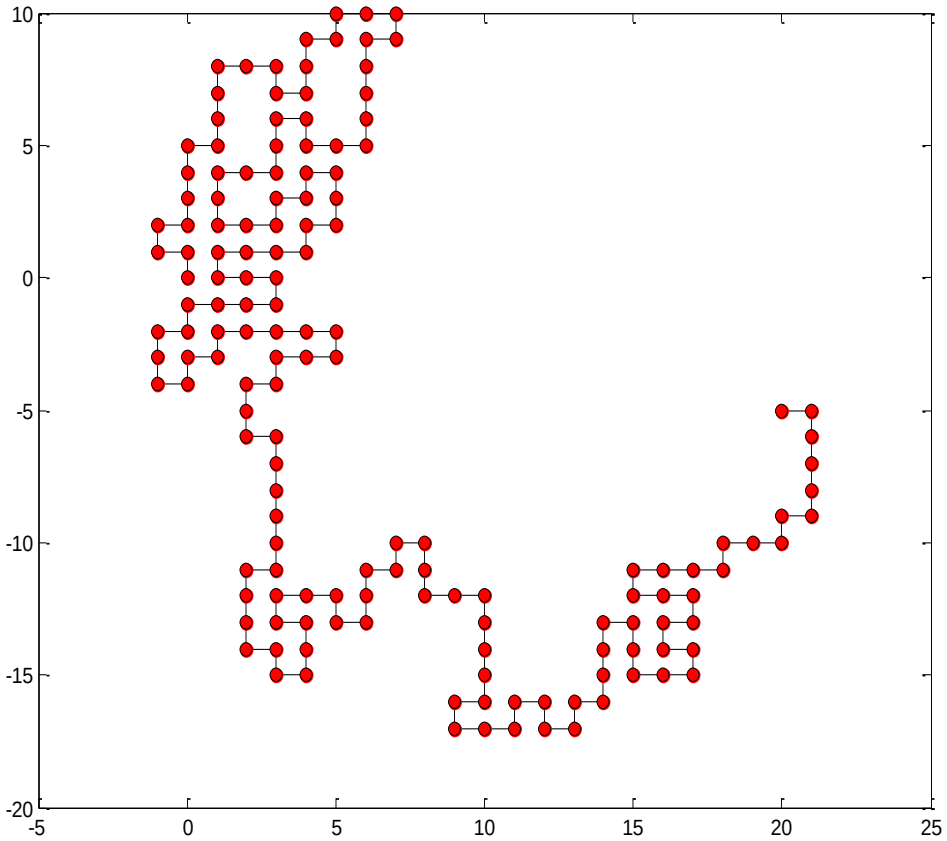
- According to this rule, the importance density function is built sequentially as

$$q_n\left(x_n \mid \boldsymbol{x}_{1:n-1}\right) = \begin{cases} 1/n_{n-1} & \text{, if } n_{n-1} > 0 \\ 0 & \text{, if } n_{n-1} = 0 \end{cases}$$

where η_{n-1} is the number of available neighbors.

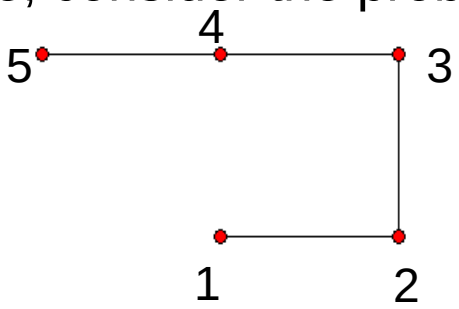
Example: Growing a Polymer in a 2D Lattice

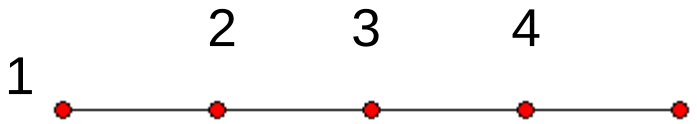
- One sample with *150* nodes (starting from (*0,0*))



Growing a Polymer in a 2D Lattice

- Notice that the chain generated in this way is not uniformly distributed.
- For example, consider the probabilities of two configurations with 5 nodes.





  - left : 1/4*1/3*1/3*1/2=1/72
  - right: 1/4*1/3*1/3*1/3=1/108
  - The importance density *q(x)* tends to generate more "compact" configurations.
- The auxiliary distributions $\gamma_n(x_{1:n})$ are just set to be the sequence of uniform distributions with *n* nodes, i.e.

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

where Z_n is the total number of chains with *n* nodes.

A C++ implementation is given here. A MatLab implementation is also available.

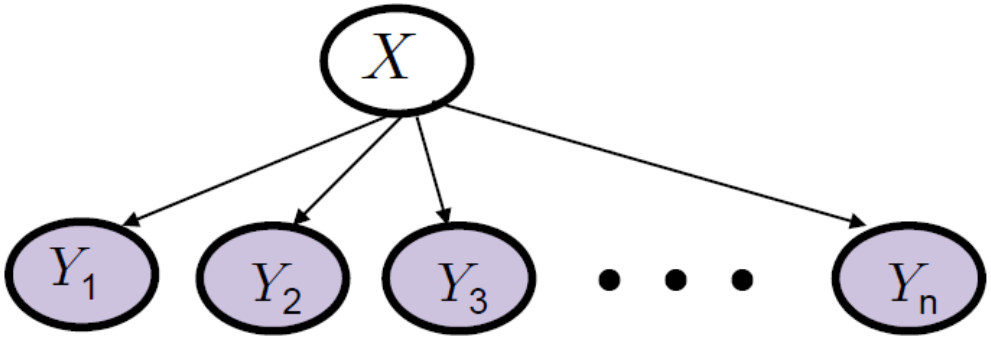
SEQUENTIAL MONTE CARLO METHODS FOR STATIC PROBLEMS

What about if all distributions π_n, $n \geq 1$ are defined on $\mathcal{X}$?

- This case can appear often, for example:
  - ✓ Sequential Bayesian Inference: $\pi_n(x) = p(x \mid \boldsymbol{y}_{1:n})$
  - ✓ Classical Bayesian Inference: $\pi_n(x) = \pi(x)$
  - ✓ Global Optimization: $\pi_n(x) \propto [\pi(x)]^{\gamma_n}$, $\gamma_n \to \infty$ increasing sequence
  - ✓ Sampling from a fixed target π : $\pi_n(x) \propto [\mu_1(x)]^{\eta_n}[\pi(x)]^{1-\eta_n}$ with μ_1 easy to sample and $\eta_1 = 1, \eta_n < \eta_{n-1},$ and $\eta_P = 0$.
  - ✓ Rare Event Simulation: $\pi(A) \ll 1$: $\pi_n(x) = \pi(x)\mathbb{I}_{E_n}(x)$, Z_1 known, $E_1 = E, E_n \subset E_{n-1}, E_P = A$, then $Z_P = \pi(A)$.
- It is not clear how SMC can be related to this problem since $\mathcal{X}_n = \mathcal{X}$ instead of $\mathcal{X}_n = \mathcal{X}^n$.

What about if all distributions π_n, n≥1 are defined on $\mathcal{X}$?

- This case can appear often, for example:
  - Sequential Bayesian Estimation: $\pi_n(x) = p(x \mid \boldsymbol{y}_{1:n})$



  - Global Optimization: $\pi_n(x) \propto \left[\pi(x)\right]^{\eta_n}, \eta_n \to \infty$
  - Sampling from a fixed distribution: $\pi_n(x) \propto \left(\mu(x)\right)^{\eta_n} \left(\pi(x)\right)^{1-\eta_n}$ where μ(x) is an easy to sample from distribution. Use a sequence

$$\eta_1 = 1 > \eta_2 > \ldots > \eta_{Final} = 0, \; i.e. \; \pi_1(x) \propto \mu(x), \; \pi_{Final}(x) \propto \pi(x)$$

What about if all distributions π_n, $n \geq 1$ are defined on $\mathcal{X}$?

- This case can appear often, for example:
  - Rare Event Simulation:

$$\pi(A) << 1 : \pi_n(x) \propto \pi_n(x)\mathbb{I}_{E_n}(x), \; Normalizing \; Factor \; Z_1 = known$$

$$Simulate \; with \; sequence : E_1 = \mathcal{X} \supset E_2 \supset \ldots \supset E_{Final} = A$$

The required probability is then: $Z_{Final} = \pi(A)$

  - The Boolean Satisfiability Problem
  - Computing Matrix Permanents
  - Computing volumes in high dimensions

Using Standard Importance Sampling

- When *n=1*, sample $X_1^{(i)} \sim q_1(.)$ to obtain $\hat{\pi}_1(x) = \sum_{i=1}^{N} W_1^{(i)} \delta_{X_1^{(i)}}(x)$

where $W_1\left(X_1^{(i)}\right) \propto w_1\left(X_1^{(i)}\right) = \frac{\gamma_1\left(X_1^{(i)}\right)}{q_1\left(X_1^{(i)}\right)}$ and $\hat{Z}_1(x) = \frac{1}{N}\sum_{i=1}^{N} w_1^{(i)}\left(X_1^{(i)}\right)$

- At step n-1 we have N particles $\left\{X_{n-1}^{(i)}, W_{n-1}^{(i)}\right\}, X_{n-1}^{(i)} \sim q_{n-1}$, and

$$W_{n-1}^{(i)} \propto \frac{\gamma_{\mathrm{n-1}}\left(X_{\mathrm{n-1}}^{(i)}\right)}{q_{\mathrm{n-1}}\left(X_{\mathrm{n-1}}^{(i)}\right)}$$

- We move the particles according to the transition kernel:

$$X_n^{(i)} \sim K_n\left(X_{n-1}^{(i)}, \cdot\right) \rightarrow q_n(x') = \int q_{n-1}(x)\, K_n(x, x')dx$$

Using Standard Importance Sampling

- The optimal kernel is $K_n(x, x') = \pi_n(x')$ which cannot be used.
- The marginal importance distribution cannot be computed

$$q_n(x_n) = \int q_{n-1}(x_{n-1})\, K_n(x_{n-1}, x_n) dx_{n-1} = \int q_1(x_1) \prod_{k=2}^{n} K_k(x_{k-1}, x_k)\, d\boldsymbol{x}_{1:n-1}$$

- Need to start with an easy to sample $\pi_1(x)$ and with small discrepancy between $\pi_1(x)$ and $q_1(x)$.
- MC approximation is $\mathcal{O}(N^2)$.

$$\hat{q}_n(x_n) = \int \hat{q}_{n-1}(x_{n-1})\, K_n(x_{n-1}, x_n) dx_{n-1} = \frac{1}{N} \sum_{i=1}^{N} K_n\left(X_{n-1}^{(i)}, x_n\right)$$

- Impossible if you need to evaluate $K_n(x_{n-1}, x_n)$ pointwise as in the MH kernel:

$$K_n(x, dx') = \alpha(x, x') q(x, dx') + \left(1 - \int a(x, u) q(x, du)\right) \delta_x(dx')$$

- In conclusion standard importance sampling cannot be used.

Transition Kernels

$$X_n^{(i)} \sim K_n\left(X_{n-1}^{(i)}, \cdot\right) \rightarrow q_n(x') = \int q_{n-1}(x)\, K_n(x, x') dx$$

- $K_n(x, x') = K_n(x')$
  - ✓ Simple parametric form (e.g. Gaussian, multinomial, etc.)
  - ✓ Semi-parametric based on $\hat{q}_{n-1}\,(dx)$

- Stavropoulos, P. and Titterington, D. M. (2001). "Improved particle filters and smoothing." In A. Doucet, N. d. F. and Gordon., N. (eds.), Sequential Monte Carlo Methods in Practice. New York, NY: Springer-Verlag
- West M. 1993. Mixtures models, Monte Carlo, Bayesian updating and dynamic models. Computer Science and Statistics 24: 325–333.

- $K_n(x, x')$ MCMC kernel of invariant distribution $\pi_n(x)$.
  - ✓ Burn-in correction by importance sampling
  - ✓ Scaling of proposal can depend on $\left\{X_{n-1}^{(i)}\right\}$

- N. Chopin, A sequential partile filter method for static problems, Biometrika, 2002, 89, 3, pp. 539-551.
- D. Crisan and A. Doucet, Convergence of Sequential MC Methods, 2002.

- $K_n(x, x')$ approximation of a Gibbs sampler of invariant distribution $\pi_n(x)$.

What about if all distributions π_n, n≥1 are defined on $\mathcal{X}$?

- Key Idea: Apply SMC to an augmented sequence of distributions $\{\tilde{\pi}_n\}_{n\geq 1}$ on $\mathcal{X}^n$:

For example:

$$\int \tilde{\pi}_n(\boldsymbol{x}_{1:n-1}, x_n) d\boldsymbol{x}_{1:n-1} = \pi_n(x_n)$$

$$\tilde{\pi}_n(\boldsymbol{x}_{1:n-1}, x_n) = \pi_n(x_n)\tilde{\pi}_n(\boldsymbol{x}_{1:n-1}|x_n)$$

$where\ \tilde{\pi}_n(\boldsymbol{x}_{1:n-1}|x_n)\ any\ conditional\ distribution\ on\ \mathcal{X}^{n-1}$

- C Jarzynski, Nonequilibrium Equality for Free Energy Differences, Phys. Rev. Lett. 78, 2690–2693 (1997)
- Gavin E. Crooks, Nonequilibrium Measurements of Free Energy Differences for Microscopically Reversible Markovian Systems, Journal of Statistical Physics March 1998, Volume 90, Issue 5-6, pp 1481-1487
- W Gilks and C. Berzuini, Following a moving target: MC inference for dynamic Bayesian Models,. JRSS B, 2001
- Neal, R. M. (2001) ``Annealed importance sampling'', Statistics and Computing, vol. 11, pp. 125-139
- N. Chopin, A sequential partile filter method for static problems, Biometrika, 2002, 89, 3, pp. 539-551.
- P. Del Moral, A. Doucet and A. Jasra, Sequential Monte Carlo for Bayesian Computation, Bayesian
- Statistics, 2006

SMC For Static Models

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

$$\pi_n(x) = \underbrace{1/Z_n}_{\text{Unknown}} \underbrace{\gamma_n(x)}_{\text{Known}}$$

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

- This is not the same as the standard SMC discussed earlier which was defined on $\mathcal{X}^n$.

$$\pi_n(\boldsymbol{x}_{1:n}) = p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$$

- We propose to work on an extended space. Let us explain this for n=2.
  - ✓ $\frac{\pi_2(x_2)}{q_2(x_2)} = \frac{\pi_2(x_2)}{\int q_1(dx_1)K_2(x_1,x_2)}$ cannot be evaluated. We propose instead to use:
  - ✓ $\frac{\pi_2(x_1,x_2)}{q_2(x_1,x_2)} = \frac{\pi_2(x_2)L_1(x_2,x_1)}{q_1(x_1)K_2(x_1,x_2)}$ where $L_1(x_2,x_1)$ is an arbitrary backward Markov kernel.
  - ✓ This is valid as $\int \pi_2(x_2)L_1(x_2,x_1)dx_1 = \pi_2(x_2)\int L_1(x_2,x_1)dx_1 = \pi_2(x_2)$ for arbitrary $L_1(x_2,x_1)$.

SMC For Static Models

- We will construct an artificial distribution that is the product of the target distribution from which we want to sample and a backward kernel *L* as follows:

$$\tilde{\pi}_n(\boldsymbol{x}_{1:n}) = Z_n^{-1}\gamma_n(\boldsymbol{x}_{1:n}),\ where:$$

$$\gamma_n(\boldsymbol{x}_{1:n}) = \gamma_n(x_n)\prod_{k=1}^{n-1} L_k(x_{k+1}, x_k)$$

$$Target: \gamma_n(x_n)$$

$$Backward\ Transitions: \prod_{k=1}^{n-1} L_k(x_{k+1}, x_k)$$

such that $\pi_n(x_n) = \int \tilde{\pi}_n(\boldsymbol{x}_{1:n}) d\boldsymbol{x}_{1:n-1}$

- The joint importance distribution is taken as:

$$q_n(\boldsymbol{x}_{1:n}) = \mu_1(x_1)\prod_{k=2}^{n} K_k(x_{k-1}, x_k) \equiv \mu_1(x_1)\prod_{k=2}^{n} q_k(x_k|x_{k-1})$$

- This is valid as you can notice that $\int \tilde{\pi}_n(\boldsymbol{x}_{1:n}) d\boldsymbol{x}_{1:n-1} = \pi_n(x_n)\int \prod_{k=1}^{n-1} L_k(x_{k+1}, x_k)\, d\boldsymbol{x}_{1:n-1} = \pi_n(x_n)$ for arbitrary $L_k(x_{k+1}, x_k)$.

SMC For Static Models

- We will construct an artificial distribution that is the product of the target distribution from which we want to sample and a backward kernel L as follows:

$$\tilde{\pi}_n(\boldsymbol{x}_{1:n}) = Z_n^{-1}\gamma_n(\boldsymbol{x}_{1:n}),\ where:$$

$$\gamma_n(\boldsymbol{x}_{1:n}) = \gamma_n(x_n)\prod_{k=1}^{n-1} L_k(x_{k+1}, x_k)$$

$$Target: \gamma_n(x_n)$$

$$Backward\ Transitions: \prod_{k=1}^{n-1} L_k(x_{k+1}, x_k)$$

such that $$\pi_n(x_n) = \int \tilde{\pi}_n(\boldsymbol{x}_{1:n})d\boldsymbol{x}_{1:n-1}$$

- The importance weights are:

$$W_n = \frac{\gamma_n(\boldsymbol{x}_{1:n})}{q_n(\boldsymbol{x}_{1:n})} = W_{n-1}\frac{q_{n-1}(\boldsymbol{x}_{1:n-1})\gamma_n(\boldsymbol{x}_{1:n})}{\gamma_{n-1}(\boldsymbol{x}_{1:n-1})q_n(\boldsymbol{x}_{1:n})} = W_{n-1}\frac{\gamma_n(x_n)\prod_{k=1}^{n-1} L_k(x_{k+1}, x_k)}{\gamma_{n-1}(x_{n-1})\prod_{k=1}^{n-2} L_k(x_{k+1}, x_k)K_n(x_{n-1}, x_n)}$$

$$= W_{n-1}\frac{\gamma_n(x_n)L_{n-1}(x_n, x_{n-1})}{\gamma_{n-1}(x_{n-1})K_n(x_{n-1}, x_n)}$$

SMC For Static Models

$$W_n = W_{n-1} \frac{\gamma_n(x_n) L_{n-1}(x_{n-1} \mid x_n)}{\gamma_{n-1}(x_{n-1}) K_n(x_{n-1}, x_n)}$$

- Any MCMC kernel can be used for the proposal $q(\cdot \mid \cdot)$ / kernel $K_k(\cdot,\cdot)$
- Since our interest is in computing only

$$\pi_n(x_n) = \int \tilde{\pi}_n(x_{1:n}) d\boldsymbol{x}_{1:n-1} = \frac{1}{Z}\gamma_n(x_n)$$

there is no degeneracy problem.

P. Del Moral, A. Doucet and A. Jasra, Sequential Monte Carlo for Bayesian Computation, Bayesian Statistics, 2006

Algorithm: SMC For Static Problems

(1) Initialize at time n=1:

(2) At time n≥2:

- Sample $\tilde{X}_n^{(i)} \sim K_n\left(X_{n-1}^{(i)}, x_n\right)$, and augment $\tilde{\boldsymbol{X}}_{n-1:n}^{(i)} \sim \left(X_{n-1}^{(i)}, \tilde{X}_n^{(i)}\right)$
- Compute the importance weights

$$W_n^{(i)} = W_{n-1}^{(i)} \frac{\gamma_n\left(\tilde{X}_n^{(i)}\right) L_{n-1}\left(\tilde{X}_{n-1}^{(i)} | \tilde{X}_n^{(i)}\right)}{\gamma_{n-1}\left(\tilde{X}_{n-1}^{(i)}\right) K_n\left(\tilde{X}_{n-1}^{(i)}, \tilde{X}_n^{(i)}\right)}$$

Then the weighted approximation is

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

- We finally resample from $X_n^{(i)} \sim \hat{\pi}_n(x_n)$ to obtain:

$$\pi_n(x_n) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_n^{(i)}}(x_n)$$

Algorithm: SMC For Static Problems

- The ratio of the normalizing constants can be computed as:

$$\frac{Z_n}{Z_{n-1}} = \frac{\int \gamma_n(x_n)dx_n}{\int \gamma_{n-1}(x_{n-1})dx_{n-1}} = \frac{\int \gamma_n(x_n)L_{n-1}(x_n, x_{n-1})dx_{n-1}dx_n}{\int \gamma_{n-1}(x_{n-1})dx_{n-1}} =$$

$$= \int \frac{\gamma_n(x_n)L_{n-1}(x_n, x_{n-1})}{\gamma_{n-1}(x_{n-1})K_n(x_{n-1}, x_n)} \pi_{n-1}(dx_{n-1})K_n(x_{n-1}, dx_n)$$

- Substituting the SMC approximation for $\hat{\pi}_{n-1}(x_{n-1}) = \sum_{i=1}^{N} W_{n-1}^{(i)} \delta_{\tilde{X}_{n-1}^{(i)}}(x_{n-1})$ and using $K_n\left(X_{n-1}^{(i)}, dx_n\right) = \delta_{X_n^{(i)}}(x_n)dx_n$, we obtain the following Monte Carlo approximation:

$$\widehat{\frac{Z_n}{Z_{n-1}}} = \sum_{i=1}^{N} W_{n-1}^{(i)} \frac{\gamma_n\left(X_n^{(i)}\right) L_{n-1}\left(X_n^{(i)}, X_{n-1}^{(i)}\right)}{\gamma_{n-1}\left(X_{n-1}^{(i)}\right) K_n\left(X_{n-1}^{(i)}, X_n^{(i)}\right)}$$

Optimal Backward Kernel

- The optimal kernel $L_{n-1}^{opt}(x_n, x_{n-1})$ is the one that brings us back to the case where there is no space extension:

$$L_{n-1}^{opt}(x_n, x_{n-1}) = \frac{q_{n-1}(x_{n-1})K_n(x_{n-1}, x_n)}{q_n(x_n)}$$

- This follows from the forward-backward formula for Markov processes

$$q_n(\boldsymbol{x}_{1:n}) = q_1(x_1)\prod_{k=2}^{n} K_k(x_{k-1}, x_k) = q_n(x_n)\prod_{k=1}^{n-1} L_{k-1}^{opt}(x_k, x_{k-1})$$

- The optimal kernel $L_{n-1}^{opt}(x_n, x_{n-1})$ can be approximated still leading to asymptotically consistent estimates.

- However, note: By extending the integration space, the variance of the importance weights can only increase.

Algorithm: SMC For Static Problems

- As in MCMC, in these calculations one can use a mixture of valid kernels, e.g. $K_n(x, x') = \sum_{m=1}^{M} \alpha_{n,m}(x) K_{n,m}(x, x')$.

- This can be implemented using an auxiliary discrete variable s.t. $Pr(M_n = m) = \alpha_{n,m}(x)$ and then performing importance sampling on the extended space.

- The incremental importance weight becomes:

$$\frac{\gamma_n(x')\beta_{n-1,m}(x')L_{n-1,\mathrm{m}}(x', x)}{\gamma_{n-1}(x)\alpha_{n,m}(x)K_{n,m}(x, x')} \quad \text{instead of} \quad \frac{\gamma_n(x')L_{n-1}(x', x)}{\gamma_{n-1}(x)K_n(x, x')}$$

- Here we have used a mixture of artificial backwards kernels as

$$L_{n-1}(x', x) = \sum_{m=1}^{M} \beta_{n-1,m}(x) L_{n-1,m}(x', x) .$$

- Optimal estimates of $\beta_{n-1,m}$, $L_{n-1,m}$ can be derived.

  - M.K. Pitt and N. Shephard, Filtering via Simulation: Auxiliary Particle Filter, JASA, 1999

Central Limit Theorem for SIS

- We have seen earlier for the standard SIS with no resampling

$$\sqrt{N}\left(\mathbb{E}_{\hat{\pi}_n}(\varphi) - \mathbb{E}_{\pi_n}(\varphi)\right) \xrightarrow{d} \mathcal{N}\left(0, \sigma^2_{IS,n}(\varphi)\right)$$

$$\sigma^2_{IS,n}(\varphi) = \int \frac{\tilde{\pi}^2_n(\boldsymbol{x}_{1:n})}{q_n(\boldsymbol{x}_{1:n})}\left(\varphi(x_n) - \mathbb{E}_{\pi_n}(\varphi)\right)^2 d\boldsymbol{x}_{1:n}$$

- With resampling at each iteration: $\sqrt{N}\left(\mathbb{E}_{\hat{\pi}_n}(\varphi) - \mathbb{E}_{\pi_n}(\varphi)\right) \xrightarrow{d} \mathcal{N}\left(0, \sigma^2_{SMC,n}(\varphi)\right)$

$$\sigma^2_{SMC,n}(\varphi) = \int \frac{\tilde{\pi}^2_n(x_1)}{q_1(x_1)}\left(\int \varphi(x_n)\tilde{\pi}_n(x_n|x_1)dx_n - \mathbb{E}_{\pi_n}(\varphi)\right)^2 dx_1$$

$$+ \sum_{k=2}^{n-1} \int \frac{\left(\tilde{\pi}_n(x_k)L_{k-1}(x_k, x_{k-1})\right)^2}{\pi_{k-1}(x_{k-1})K_k\left(x_{k-1}, x_k\right)}\left(\int \varphi(x_n)\tilde{\pi}_n(x_n|x_k)dx_n - \mathbb{E}_{\pi_n}(\varphi)\right)^2 d\boldsymbol{x}_{k-1:k}$$

$$+ \int \frac{\left(\pi_n(x_n)L_{n-1}(x_n, x_{n-1})\right)^2}{\pi_{n-1}(x_{n-1})K_n\left(x_{n-1}, x_n\right)}\left(\varphi(x_n) - \mathbb{E}_{\pi_n}(\varphi)\right)^2 d\boldsymbol{x}_{n-1:n}$$

- Under mixing assumptions $\sigma_{SMC,n}(\varphi)$ is upper bounded over time.
  - P. Del Moral, Feynman-Kac Formulae, Chapter 7, Springer-Verlag, 2004
  - Del Moral, P., Doucet A., Peters G.W. Sharp Propagations of Chaos Estimates for Feynman-Kac particle Models (preliminary version) Probability Theory and its Applications, SIAM, vol. 51, no. 3, pp. 459--485 (2006).

Asymptotic Bias

- We have seen earlier for the standard SIS with no resampling

$$N\left(\mathbb{E}_{\hat{\pi}_n}(\varphi) - \mathbb{E}_{\pi_n}(\varphi)\right) \xrightarrow{d} -\int \frac{\pi_n^2(\boldsymbol{x}_{1:n})}{q_n(\boldsymbol{x}_{1:n})}\left(\varphi(x_n) - \mathbb{E}_{\pi_n}(\varphi)\right) d\boldsymbol{x}_{1:n}$$

- With resampling at each iteration:

$$N\left(\mathbb{E}_{\hat{\pi}_n}(\varphi) - \mathbb{E}_{\pi_n}(\varphi)\right) \xrightarrow{d} b_{\mathrm{SMC},n}(\varphi)$$

$$b_{SMC,n}(\varphi) = -\int \frac{\tilde{\pi}_n^2(x_1)}{q_1(x_1)}\left(\int \varphi(x_n)\tilde{\pi}_n(x_n|x_1)dx_n - \mathbb{E}_{\pi_n}(\varphi)\right)dx_1$$

$$-\sum_{k=2}^{n-1}\int \frac{\left(\tilde{\pi}_n(x_k)L_{k-1}(x_k, x_{k-1})\right)^2}{\pi_{k-1}(x_{k-1})K_k(x_{k-1,}x_k)}\left(\int \varphi(x_n)\tilde{\pi}_n(x_n|x_k)dx_n - \mathbb{E}_{\pi_n}(\varphi)\right)d\boldsymbol{x}_{k-1:k}$$

$$-\int \frac{\left(\pi_n(x_n)L_{n-1}(x_n, x_{n-1})\right)^2}{\pi_{n-1}(x_{\mathrm{n}-1})K_n(x_{n-1,}x_n)}\left(\varphi(x_n) - \mathbb{E}_{\pi_n}(\varphi)\right)d\boldsymbol{x}_{n-1:n}$$

- Under mixing assumptions $\left|b_{SMC,n}(\varphi)\right|$ is upper bounded over time.

- P. Del Moral, Feynman-Kac Formulae, Chapter 7, Springer-Verlag, 2004
- Del Moral, P., Doucet A., Peters G.W. Sharp Propagations of Chaos Estimates for Feynman-Kac particle Models (preliminary version) Probability Theory and its Applications, SIAM, vol. 51, no. 3, pp. 459--485 (2006).

SMC For Static Models: Choice of L

- A default choice is first using a π_n-invariant MCMC kernel q_n and then the corresponding reversed kernel L_n approximating L_n^{opt}:

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

- Using this easy choice, we can simplify the expression for the weights:

$$W_n = W_{n-1}\frac{\gamma_n(x_n)L_{n-1}(x_n, x_{n-1})}{\gamma_{n-1}(x_{n-1})K_n(x_{n-1}, x_n)} = W_{n-1}\frac{\gamma_n(x_n)}{\gamma_{n-1}(x_{n-1})K_n(x_{n-1}, x_n)}\frac{\pi_n(x_{n-1})K_n(x_{n-1}, x_n)}{\pi_n(x_n)} \Rightarrow$$

$$W_n = W_{n-1}\frac{\gamma_n(X_{n-1}^{(i)})}{\gamma_{n-1}(X_{n-1}^{(i)})}$$

- This is known as *annealed importance sampling*. The particular choice has been used in physics and statistics.

W Gilks and C. Berzuini, Following a moving target: MC inference for dynamic Bayesian Models, JRSS B, 2001

ONLINE PARAMETER ESTIMATION

Online Bayesian Parameter Estimation

- Assume that our state model is defined with some unknown static parameter θ with some prior $p(\theta)$:

$$X_1 \sim \mu(.) \; and \; X_n \,|\, (X_{n-1} = x_{n-1}) \sim f_\theta(x_n \,|\, x_{n-1})$$

$$Y_n \,|\, (X_n = x_n) \sim g_\theta(y_n \,|\, x_n)$$

- Given data $\boldsymbol{y}_{1:n}$, inference now is based on:

$$p(\theta, \boldsymbol{x}_{1:n} \,|\, \boldsymbol{y}_{1:n}) = p(\theta \,|\, \boldsymbol{y}_{1:n})\, p_\theta(\boldsymbol{x}_{1:n} \,|\, \boldsymbol{y}_{1:n}),$$

$$where$$

$$p(\theta \,|\, \boldsymbol{y}_{1:n}) \propto p_\theta(\boldsymbol{y}_{1:n})\, p(\theta)$$

- We can use standard SMC but on the extended space $Z_n=(X_n,\theta_n)$.

$$f(z_n \,|\, z_{n-1}) = \delta_{\theta_{n-1}}(\theta_n)\, f_\theta(x_n \,|\, x_{n-1}), \; g(y_n \,|\, z_n) = g_\theta(y_n \,|\, x_n)$$

- Note that θ is a static parameter –does not involve with n.

Online Bayesian Parameter Estimation

- For fixed θ, using our earlier error estimates

$$Var[\log \hat{p}_\theta(\boldsymbol{y}_{1:n})] \leq \frac{Cn}{N}$$

- In a Bayesian context, the problem is even more severe as

$$p(\theta \mid \boldsymbol{y}_{1:n}) \propto p_\theta(\boldsymbol{y}_{1:n})\, p(\theta)$$

- Exponential stability assumption cannot hold as $\theta_n = \theta_1$.

- To mitigate this problem, introduce MCMC steps on θ.

- C. Andrieu, N. De Freitas and A. Doucet, Sequential MCMC for Bayesian Model Selection, Proc. IEEE Workshop HOS, 1999
- P. Fearnhead, MCMC, sufficient statistics and particle filters, JCGS, 2002
- W Gilks and C. Berzuini, Following a moving target: MC inference for dynamic Bayesian Models, JRSS B, 2001
- Storvik, G., 2002, Particle filters in state space models with the presence of unknown static parameters, IEEE. Trans. of Signal Processing 50, 281—289.
- Eric Jacquier, Michael Johannes, Nicholas Polson, MCMC maximum likelihood for latent state models, In Journal of Econometrics, Volume 137, Issue 2, 2007, Pages 615-640.
- T. Vercauteren et al, Batch and Sequential Bayesian Estimators of the Number of Active Terminals in an IEEE 802.11 Network.

Recursive Bayesian Parameter Estimation using SMC

- We set $\theta \sim \pi(\theta)$, and try to estimate $p(\theta, \boldsymbol{x}_{1:n} | \boldsymbol{y}_{1:n})$ using sequential Monte Carlo.
- SMC is applicable as we know $p(\theta, \boldsymbol{x}_{1:n} | \boldsymbol{y}_{1:n})$ up to normalizing factor.
- However, SMC does not work as we only sample particles in Θ space at time $n = 1$ and never modify their locations. Thus after a few time steps, $p(\theta \, | \boldsymbol{y}_{1:n})$ is approximated by a single particle.
- The higher the dimensionality of θ, the faster the degeneracy arises.
- A potential solution is by adding artificial noise, e.g. $\theta_k = \theta_{k-1} + \varepsilon_k$. This modifies however the target distributions.

Online Bayesian Parameter Estimation

- When

$$p(\theta|\boldsymbol{y}_{1:n}, \boldsymbol{x}_{1:n}) = p\big(\theta|s_n(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n})\big)$$

where $s_n(x_{1:n}, y_{1:n})$ is finite-dimensional, this becomes an elegant algorithm that however still has the degeneracy problem since it uses $\hat{p}(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$

- As dim(Z_n) = dim(X_n) + dim(θ), such methods are not recommended for high-dimensional θ, especially with vague priors.

Artificial Dynamics for θ

- A solution consists of perturbing the location of the particles $\{\theta^{(i)}\}$ in a way that does not modify their distributions; i.e. if at time n

$$\theta_n^{(i)} \sim p(\theta \mid \boldsymbol{y}_{1:n})$$

then we would like a transition kernel such that if

$$\theta_n^{'(i)} \mid \theta_n^{(i)} \sim M_n(\theta_n^{(i)}, .)$$

Then:

$$\theta_n^{'(i)} \sim p(\theta \mid \boldsymbol{y}_{1:n})$$

- In Markov chain language, we want $M_n(\theta, \theta')$ to be $p(\theta \mid \boldsymbol{y}_{1:n})$ invariant.

Artificial Dynamics Using MCMC

- There is a whole literature on the design of such kernels known as Markov chain Monte Carlo e.g. the Metropolis-Hastings algorithm.

- We cannot use these algorithms directly as $p\left(\theta \mid \boldsymbol{y}_{1:n}\right)$ would need to be known up to a normalizing constant but $p\left(\boldsymbol{y}_{1:n} \mid \theta\right) \equiv p_{\theta}\left(\boldsymbol{y}_{1:n}\right)$ is unknown.

- However, we can use a simple Gibbs update.

$$\theta_n^{(i)} \sim p\left(\theta \mid \boldsymbol{y}_{1:n}, \boldsymbol{X}_{1:n}^{(i)}\right)$$

- Indeed note that if $\left(\boldsymbol{X}_{1:n}^{(i)}, \theta_n^{(i)}\right) \sim p\left(\theta, \boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)$ then if $\theta_n^{'(i)} \sim p\left(\theta \mid \boldsymbol{y}_{1:n}, \boldsymbol{X}_{1:n}^{(i)}\right)$, we have:

$$\left(\boldsymbol{X}_{1:n}^{(i)}, \theta_n^{'(i)}\right) \sim p\left(\theta, \boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)$$

- Indeed note that:

$$\int p\left(\theta, \boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right) p\left(\theta' \mid \boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}\right) d\theta = p\left(\theta', \boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)$$

SMC with MCMC for Parameter Estimation

- Given an approximation at time n-1:

$$\hat{p}(\theta, \boldsymbol{x}_{1:n-1}|\boldsymbol{y}_{1:n-1}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\left(\theta_{n-1}^{(i)}, \boldsymbol{X}_{1:n-1}^{(i)}\right)}(\theta, \boldsymbol{x}_{1:n-1})$$

- Sample $\tilde{X}_n^{(i)} \sim f_{\theta_{n-1}^{(i)}}\left(x_n|X_{n-1}^{(i)}\right)$, set $\tilde{X}_{1:n}^{(i)} \sim \left(\boldsymbol{X}_{1:n-1}^{(i)}, \tilde{X}_n^{(i)}\right)$ and then approximate:

$$\tilde{p}(\theta, \boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) = \sum_{i=1}^{N} W_n^{(i)} \delta_{\left(\theta_{n-1}^{(i)}, \tilde{\boldsymbol{X}}_{1:n}^{(i)}\right)}(\boldsymbol{x}_{1:n}),\ W_n^{(i)} \propto g_{\theta_{n-1}^{(i)}}\left(y_n|\tilde{X}_n^{(i)}\right)$$

- Resample $\boldsymbol{X}_{1:n}^{(i)} \sim \tilde{p}(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$, then sample $\theta_n^{(i)} \sim p\left(\theta|\boldsymbol{y}_{1:n}, \boldsymbol{X}_{1:n}^{(i)}\right)$ to obtain

$$\hat{p}(\theta, \boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\left(\theta_n^{(i)}, \boldsymbol{X}_{1:n}^{(i)}\right)}(\theta, \boldsymbol{x}_{1:n})$$

SMC with MCMC for Parameter Estimation

- Consider the following model:

$$X_{n+1} = \theta X_n + \sigma_v V_{n+1}, V_n \sim \mathcal{N}(0,1)$$
$$Y_n = X_n + \sigma_w W_n, W_n \sim \mathcal{N}(0,1)$$
$$X_1 \sim \mathcal{N}(0, \sigma_0^2)$$

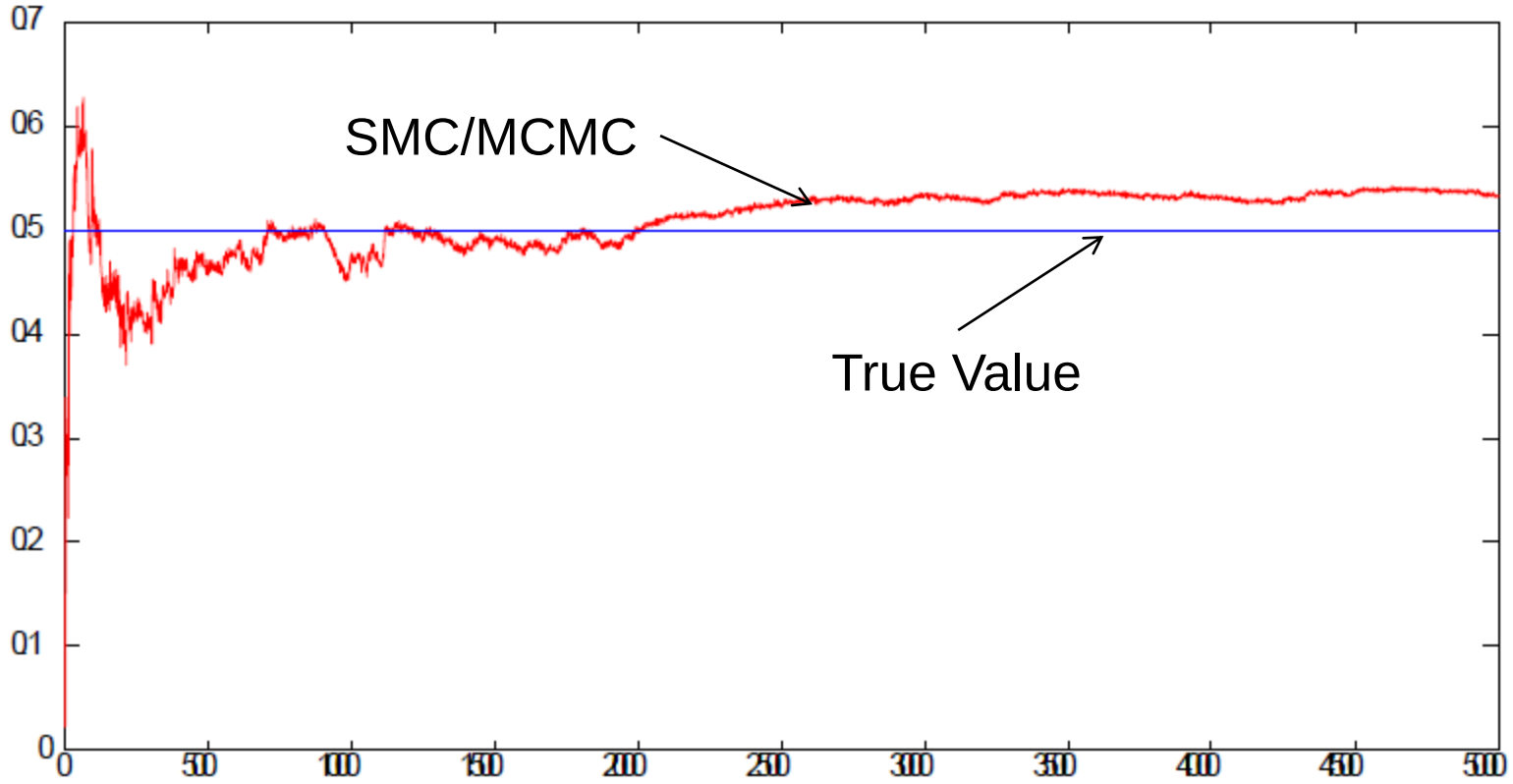
- We set the prior on θ as $\theta \sim \mathcal{U}(-1,1)$.

- In this case,

$$p(\theta \mid \boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}) \propto \mathcal{N}(\theta \mid m_n, \sigma_n^2) \mathbb{I}_{(-1,1)}(\theta)$$
$$m_n = \sigma_n^2 \left(\sum_{k=2}^{n} x_k x_{k-1} \right), \sigma_n^{-2} = \sum_{k=2}^{n-1} x_k^2$$

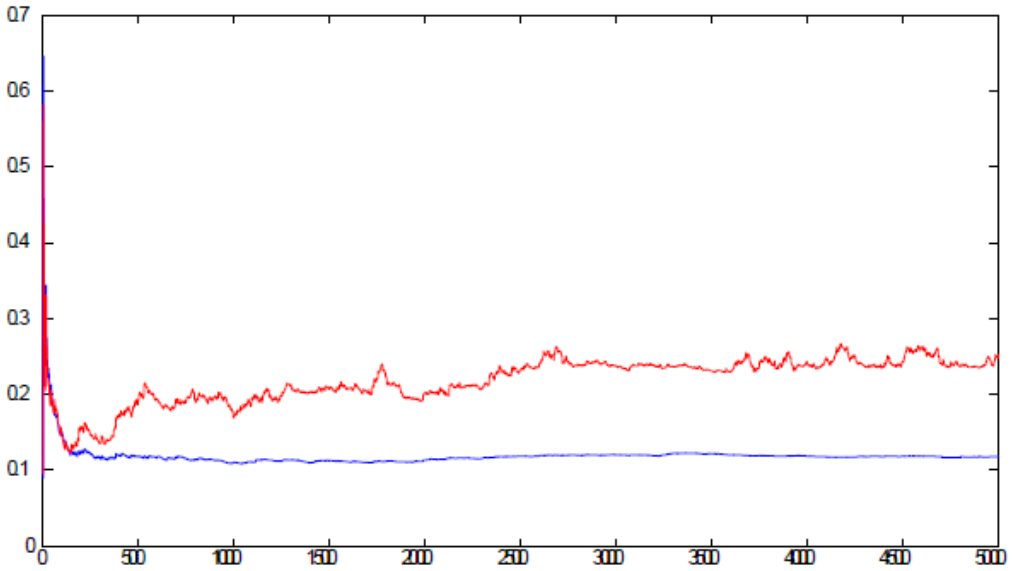
SMC with MCMC for Parameter Estimation

- We use SMC with Gibbs step. The degeneracy problem remains. SMC estimate is shown of $\mathbb{E}[\theta|\mathbf{y}_{1:n}]$ as n increases (From A. Doucet, lecture notes). The parameter converges to the wrong value.



SMC with MCMC for Parameter Estimation

- The problem with this approach is that although we move θ according to $p(\theta \mid \boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n})$, we are still relying implicitly on the approximation of the joint distribution $p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$
- As n increases, the SMC approximation of $p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$ deteriorates and the algorithm cannot converge towards the right solution. If it does converge, the results may be incorrect.



Sufficient statistics $\frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[X_k^2 \mid \boldsymbol{y}_{1:n}\right]$ computed exactly through the Kalman filter (blue) vs SMC approximation (red) for a fixed value of θ.

SMC with MCMC for Parameter Estimation

- Given data $\boldsymbol{y}_{1:n}$, inference relies on

$$p(\theta, \boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}) = p(\theta \mid \boldsymbol{y}_{1:n})\, p_\theta(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$$

where

$$p(\theta \mid \boldsymbol{y}_{1:n}) = p_\theta(\boldsymbol{y}_{1:n})\, p(\theta)$$

- We have seen that SMC are rather inefficient to sample from $p(\theta \mid \boldsymbol{y}_{1:n})$ so we look here at an MCMC approach.

- For a given parameter value θ, SMC can estimate both

$$p_\theta(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}) \; and \; p_\theta(\boldsymbol{y}_{1:n})$$

- It will be useful if we can use SMC within MCMC to sample from

$$p(\theta, \boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$$

- C. Andrieu, R. Holenstein and G.O. Roberts, Particle Markov chain Monte Carlo methods, J. R. Statist. Soc.B (2010) 72, Part 3, pp. 269–342

Gibbs Sampling Strategy

- Using Gibbs Sampling, we can sample iteratively from $p\left(\theta \mid \boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}\right)$ and $p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}, \theta\right)$
- However, it is impossible to sample from $p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}, \theta\right)$
- We can sample from $p\left(x_k \mid \boldsymbol{y}_{1:n}, \theta, \boldsymbol{x}_{1:k-1}, \boldsymbol{x}_{k+1:n}\right)$ instead but convergence will be slow.
- Alternatively, we would like to use Metropolis Hastings step to sample from $p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}, \theta\right)$, i.e. sample $\boldsymbol{X}_{1:n}^{*} \sim q\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)$ and accept with probability

$$\min\left(1, \frac{p\left(\boldsymbol{X}_{1:n}^{*} \mid \boldsymbol{y}_{1:n}, \theta\right) q\left(\boldsymbol{X}_{1:n} \mid \boldsymbol{y}_{1:n}\right)}{p\left(\boldsymbol{X}_{1:n} \mid \boldsymbol{y}_{1:n}, \theta\right) q\left(\boldsymbol{X}_{1:n}^{*} \mid \boldsymbol{y}_{1:n}\right)}\right)$$

Gibbs Sampling Strategy

- We will use the output of an SMC method approximating $p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}, \theta\right)$ as a proposal distribution.

- We know that the SMC approximation $\hat{p}(\boldsymbol{x}_{1:n} | \boldsymbol{y}_{1:n}, \theta)$ degrades an n increases but we also have under mixing assumptions:

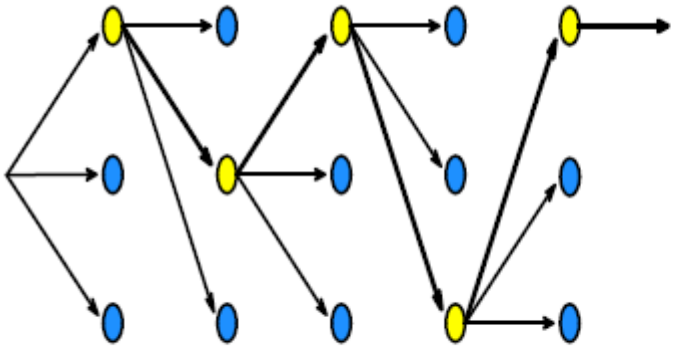
$$\left\| \mathcal{L}aw(\boldsymbol{X}_{1:n}^{(i)}) - p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}, \theta\right) \right\|_{TV} \leq \frac{Cn}{N}$$

  Thus things degrade linearly rather than exponentially.

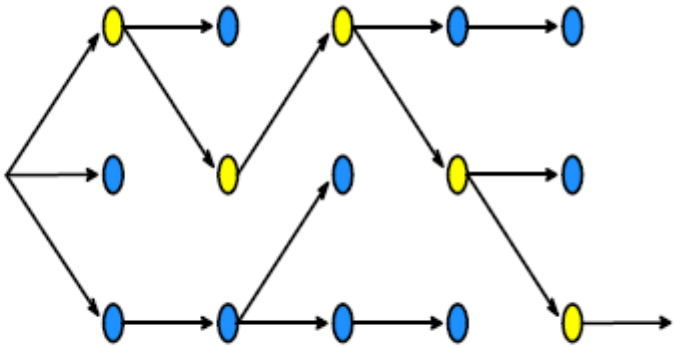
- These results are useful for small C.

Configurational Biased Monte Carlo

- The idea is related to that in the Configurational Biased MC Method (CBMC) in Molecular simulation. There are however significant differences.



CBMC for $N = 3$ particles



SMC for $N = 3$ particles

- CBMC looks like an SMC method but at each step only one particle is selected that gives N offsprings. Thus if you have done the wrong selection, then you cannot recover later on.

- D Frenkel, G C A M Mooij and B Smit, Novel scheme to study structural and thermal properties of continuously deformable molecules, I. Phys.: Condens. Matter 3 (1991) 3053-3076.

MCMC

- We use as proposal distribution $\boldsymbol{X}_{1:n}^{*} \sim \hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}, \theta)$ and store the estimate $\hat{p}_N(\boldsymbol{y}_{1:n}|\theta)$

- It is impossible to compute analytically the unconditional law of a particle as it would require integrating out $(N-1)n$ variables.

- The solution inspired by CBMC is as follows: For the current state of the Markov chain $\boldsymbol{X}_{1:n}$, run a virtual SMC method with only N-1 free particles. Ensure that at each time step k, the current state X_k is deterministically selected and compute the associated estimate $\tilde{p}_N(\boldsymbol{y}_{1:n}|\theta)$. Finally accept $\boldsymbol{X}_{1:n}^{*}$ with probability

$$\min\left(1, \frac{\hat{p}_N(\boldsymbol{y}_{1:n}|\theta)}{\tilde{p}_N(\boldsymbol{y}_{1:n}|\theta)}\right)$$

Otherwise stay where you are.

- D Frenkel, G C A M Mooij and B Smit, Novel scheme to study structural and thermal properties of continuously deformable molecules, I. Phys.: Condens. Matter 3 (1991) 3053-3076.

MCMC

- It can be shown that the algorithm defines a valid MCMC move of invariant distribution $p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n},\theta)$

- In comparison to CBMC, this algorithm enjoys the properties that the acceptance rate goes to 1 as $N \to \infty$ as

$$\hat{p}_N(\boldsymbol{y}_{1:n}|\theta) \to p(\boldsymbol{y}_{1:n}|\theta) \text{ and } \tilde{p}_N(\boldsymbol{y}_{1:n}|\theta) \to p(\boldsymbol{y}_{1:n}|\theta)$$

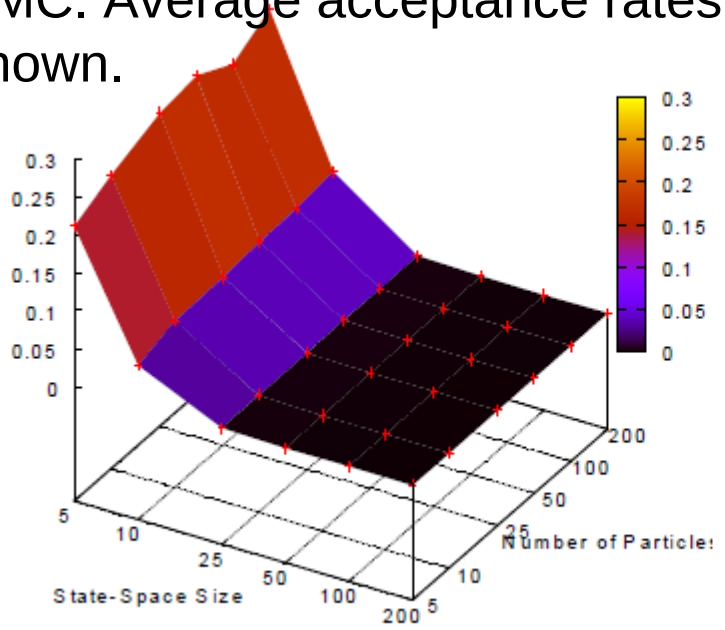
Example

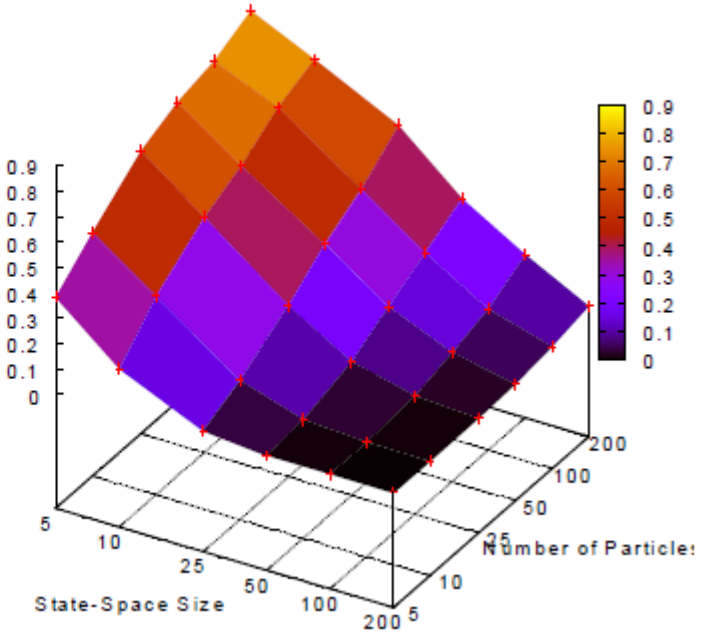
- Consider the following model:

$$X_k = \frac{1}{2} X_{k-1} + 25 \frac{X_{k-1}}{1 + X_{k-1}^2} + 8\cos(1.2k) + V_k, V_k \sim \mathcal{N}(0,15)$$

$$Y_n = \frac{X_k^2}{2} + W_k, W_k \sim \mathcal{N}(0,0.01), X_1 \sim \mathcal{N}(0,5)$$

- We take the same prior proposal distribution for both CBMC and SMC. Average acceptance rates for CBMC (left) and SMC (right) are shown.





Example

- We now consider the case when both variances σ_v^2, σ_w^2 are unknown and we take inverse Gamma (conditionally conjugate) vague priors.

- We sample from $p\left(\boldsymbol{x}_{1:500}, \theta \mid \boldsymbol{y}_{1:500}\right)$ using two strategies
  - The MCMC algorithm with N=1000 and the local EKF proposal as importance distribution. Average acceptance rate 0.43.
  - An algorithm updating the components one at a time using an MH step

    of invariance distribution $p\left(x_k \mid x_{k-1}, x_{k+1}, y_k\right)$ with the same proposal.

- In both cases we update the parameters according to $p\left(\theta \mid \boldsymbol{x}_{1:500}, \boldsymbol{y}_{1:500}\right)$

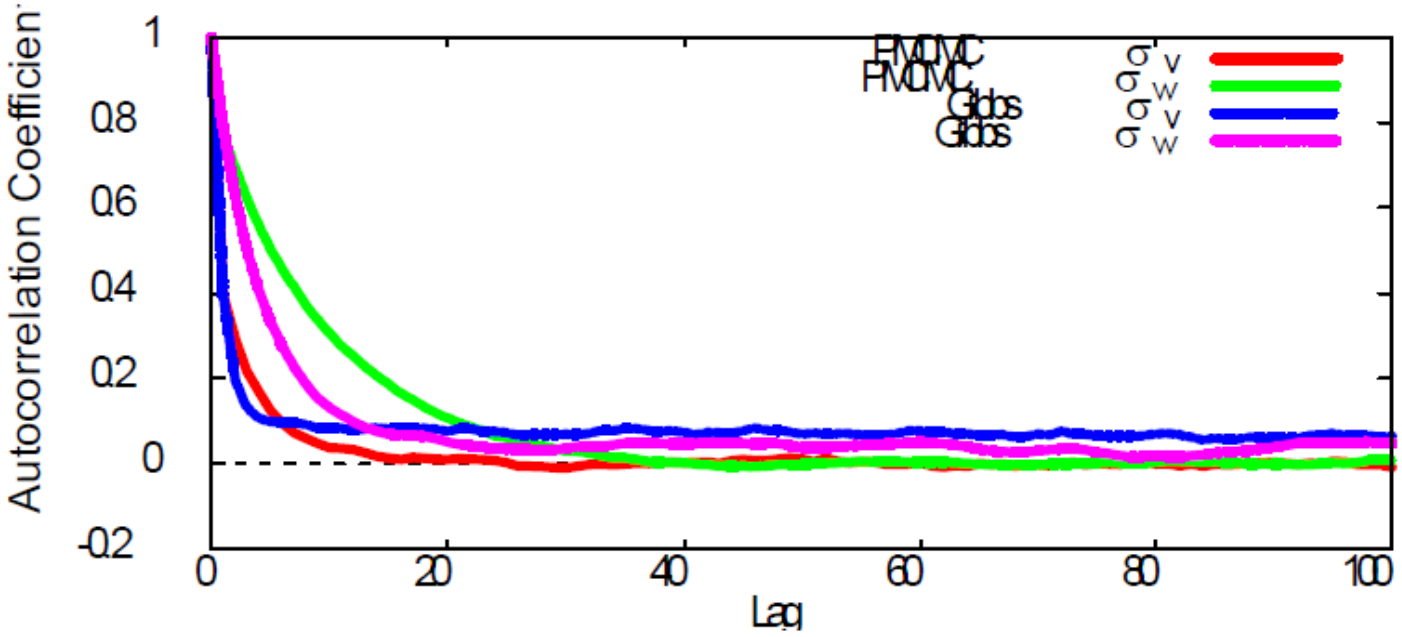
- Both algorithms are run with the same computational effort.

Autocorrelations of Simulated Markov Chains

- MH one at a time missed the mode and estimates $\mathbb{E}\left[\sigma_v^2 \mid \boldsymbol{y}_{1:500}\right] \approx 13.7\ (MH)$

$$\mathbb{E}\left[\sigma_v^2 \mid \boldsymbol{y}_{1:500}\right] \approx 9.9\ (PF-MCMC)$$

$$\sigma_v^2 = 10.$$



- If $\mathbf{X}_{1:n}$ and θ are very correlated, the MCMC algorithm is very inefficient. We will next discuss the Marginal Metropolis Hastings.

Review of Metropolis Hastings Algorithm

- As a review of MH, the proposal distribution is a Markov Chain with kernel density $q(x,y)$=q(y|x) and target distribution $\pi(x)$.

Algorithm: Generic Metropolis Hastings Sampler

- Initialization: Choose an arbitrary starting value x^0
- Iteration t $(t \geq 1)$

1. Given x^{t-1}, generate $\tilde{x} \sim q(x^{(t-1)}, x)$
2. Compute:

$$\rho\left(x^{(t-1)}, \tilde{x}\right) = \min\left(1, \frac{\pi(\tilde{x}) / q(x^{(t-1)}, \tilde{x})}{\pi\left(x^{(t-1)}\right) / q(\tilde{x}, x^{t-1})}\right)$$

3. With probability $\rho(x^{t-1}, \tilde{x})$, accept $\tilde{x}$ and set $x^t = \tilde{x}$;
   Otherwise reject $\tilde{x}$ and set $x^t = x^{t-1}$.

- It can be easily shown that $\pi(x') = \int \pi(x) \underbrace{K(x, x')}_{\text{Transition Kernel}} dx$ and under weak assumptions: $X^{(i)} \sim \pi(x)$ as $i \to \infty$

Marginal Metropolis Hastings Algorithm

- Consider the target:

$$p\left(\theta, \boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right) = p\left(\theta \mid \boldsymbol{y}_{1:n}\right) p_{\theta}\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)$$

- We use the following proposal distribution:

$$q\left(\left(\boldsymbol{x}_{1:n}^{*}, \theta^{*}\right) \mid \left(\boldsymbol{x}_{1:n}, \theta\right)\right) = q\left(\theta^{*} \mid \theta\right) p_{\theta^{*}}\left(\boldsymbol{x}_{1:n}^{*} \mid \boldsymbol{y}_{1:n}\right)$$

- Then the acceptance probability becomes:

$$\min\left(1, \frac{p\left(\theta^{*}, \boldsymbol{x}_{1:n}^{*} \mid \boldsymbol{y}_{1:n}\right) q\left(\left(\boldsymbol{x}_{1:n}, \theta\right) \mid \left(\boldsymbol{x}_{1:n}^{*}, \theta^{*}\right)\right)}{p\left(\theta, \boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right) q\left(\left(\boldsymbol{x}_{1:n}^{*}, \theta^{*}\right) \mid \left(\boldsymbol{x}_{1:n}, \theta\right)\right)}\right) =$$

$$\min\left(1, \frac{p_{\theta^{*}}\left(\boldsymbol{y}_{1:n}\right) p\left(\theta^{*}\right) q\left(\theta \mid \theta*\right)}{p_{\theta}\left(\boldsymbol{y}_{1:n}\right) p\left(\theta\right) q\left(\theta^{*} \mid \theta\right)}\right)$$

- We will use SMC approximations to compute

$$p_{\theta}\left(\boldsymbol{y}_{1:n}\right), \text{and } p_{\theta}\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)$$

Marginal Metropolis Hastings Algorithm

- Step 1:

$$Given: \left\{\theta^{(i-1)}, \boldsymbol{X}_{1:n}^{(i-1)}, \hat{p}_{\theta^{(i-1)}}(\boldsymbol{y}_{1:n})\right\}$$
$$Sample: \theta^* \sim q\left(\theta|\theta^{(i-1)}\right)$$
$$Run\ an\ SMC\ to\ obtain: \hat{p}_{\theta^*}(\boldsymbol{y}_{1:n}), p_{\theta^*}(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$$

- Step 2:

$$Sample: \boldsymbol{X}_{1:n}^* \sim \hat{p}_{\theta^*}(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$$

- Step 3: With probability $\min\left(1, \frac{\hat{p}_{\theta^*}(\boldsymbol{y}_{1:n})p(\theta^*)q\left(\theta^{(i-1)}|\theta^*\right)}{\hat{p}_{\theta^{(i-1)}}(\boldsymbol{y}_{1:n})p(\theta^{(i-1)})q(\theta^*|\theta^{(i-1)})}\right)$

$$Set: \left\{\theta^{(i)}, \boldsymbol{X}_{1:n}^{(i)}, \hat{p}_{\theta^{(i)}}(\boldsymbol{y}_{1:n})\right\} = \{\theta^*, \boldsymbol{X}_{1:n}^*, \hat{p}_{\theta^*}(\boldsymbol{y}_{1:n})\}$$
$$Otherwise: \left\{\theta^{(i)}, \boldsymbol{X}_{1:n}^{(i)}, \hat{p}_{\theta^{(i)}}(\boldsymbol{y}_{1:n})\right\} = \left\{\theta^{(i-1)}, \boldsymbol{X}_{1:n}^{(i-1)}, \hat{p}_{\theta^{(i-1)}}(\boldsymbol{y}_{1:n})\right\}$$

- The advantage of SMC is that it builds automatically efficient very high-dimensional proposal distributions based only on low-dimensional proposals.

Marginal Metropolis Hastings Algorithm

- This algorithm (without sampling $\mathbf{X}_{1:n}$) was proposed as an approximate MCMC algorithm to sample from p ($\theta \mid \mathbf{y}_{1:n}$) in (Fernandez-Villaverde & Rubio-Ramirez, 2007).
- When N≥1, the algorithm admits exactly p ($\theta, \mathbf{x}_{1:n} \mid \mathbf{y}_{1:n}$) as invariant distribution (Andrieu, D. & Holenstein, 2010). A particle version of the Gibbs sampler also exists.
- The higher N, the better the performance of the algorithm: N scales roughly linearly with n.
- Useful when X_n is moderate dimensional & θ high dimensional. Admits the plug and play property (Ionides et al., 2006).

- Jesus Fernandez-Villaverde & Juan F. Rubio-Ramirez, 2007. "On the solution of the growth model with investment-specific technological change," Applied Economics Letters, 14(8), pages 549-553.
- C Andrieu, A Doucet, R Holenstein, Particle Markov chain Monte Carlo methods, Journal of the Royal Statistical Society: Series B (Statistical Methodology) Volume 72, Issue 3, pages 269–342, June 2010
- IONIDES, E. L., BRETO, C. AND KING, A. A. (2006). Inference for nonlinear dynamical systems. *Proceedings of the Nat Acad of Sciences* **103** 18438-18443. doi. Supporting online material. Pdf and supporting text.
- R.J. Boys, D.J. Wilkinson, T. B.L. Kirkwood, Bayesian inference for a discretely observed stochastic kinetic model, J Statistics and Computing 2008, 8 June, 1573-1375.

Recursive Parameter Estimation

Recursive MLE Parameter Estimation

- The log likelihood can be written as:

$$\ell_n(\theta) = \log p_\theta(\boldsymbol{Y}_{1:n}) = \sum_{k=1}^{n} \log p_\theta\left(Y_k \mid \boldsymbol{Y}_{1:k-1}\right)$$

- Here we compute:

$$p_\theta\left(Y_k \mid \boldsymbol{Y}_{1:k-1}\right) = \int g_\theta\left(Y_k \mid x_k\right) p_\theta\left(x_k \mid \boldsymbol{Y}_{1:k-1}\right) dx_k$$

- Under regularity assumptions $\left\{X_n, Y_n, p_\theta\left(x_n \mid \boldsymbol{Y}_{1:n-1}\right)\right\}$ is an homogeneous Markov chain whish converges towards its invariant distribution:

$$\lim_{n\to\infty} \frac{\ell_n(\theta)}{n} = \ell(\theta) = \int \log \int g_\theta(y \mid x)\mu(dx)\lambda_{\theta,\theta^*}(dy, d\mu)$$

Robbins-Monro Algorithm for MLE

- We can maximize $\ell(\theta)$ by using the gradient and the Robbins-Monro algorithm:

$$\theta_n = \theta_{n-1} + \gamma_n \nabla \log p_{\theta_{1:n-1}}(Y_n \mid \mathbf{Y}_{1:n-1})$$

where:

$$\nabla p_\theta(Y_n \mid \mathbf{Y}_{1:n-1}) = \int \nabla g_\theta\left(Y_n \mid x_n\right) p_\theta\left(x_n \mid \mathbf{Y}_{1:n-1}\right) dx_n + \int g_\theta\left(Y_n \mid x_n\right) \nabla p_\theta\left(x_n \mid \mathbf{Y}_{1:n-1}\right) dx_n$$

- We thus need to approximate the signed measures: $\left\{\nabla p_\theta\left(x_n \mid \mathbf{Y}_{1:n-1}\right)\right\}$

Importance Sampling Estimation of Sensitivity

- The various proposed approximations are based on the identity:

$$\nabla p_\theta(x_n \mid \boldsymbol{Y}_{1:n-1}) = \int \frac{\nabla p_\theta(\boldsymbol{x}_{1:n} \mid \boldsymbol{Y}_{1:n-1})}{p_\theta(\boldsymbol{x}_{1:n} \mid \boldsymbol{Y}_{1:n-1})} p_\theta(\boldsymbol{x}_{1:n} \mid \boldsymbol{Y}_{1:n-1}) d\boldsymbol{x}_{1:n-1}$$

$$= \int \nabla \log p_\theta(\boldsymbol{x}_{1:n} \mid \boldsymbol{Y}_{1:n-1}) p_\theta(\boldsymbol{x}_{1:n} \mid \boldsymbol{Y}_{1:n-1}) d\boldsymbol{x}_{1:n-1}$$

- We thus can use a SMC approximation of the form:

$$\widehat{\nabla p_\theta}(x_n \mid \boldsymbol{Y}_{1:n-1}) = \frac{1}{N} \sum_{i=1}^{N} \alpha_n^{(i)} \delta_{X_n^{(i)}}(x_n)$$

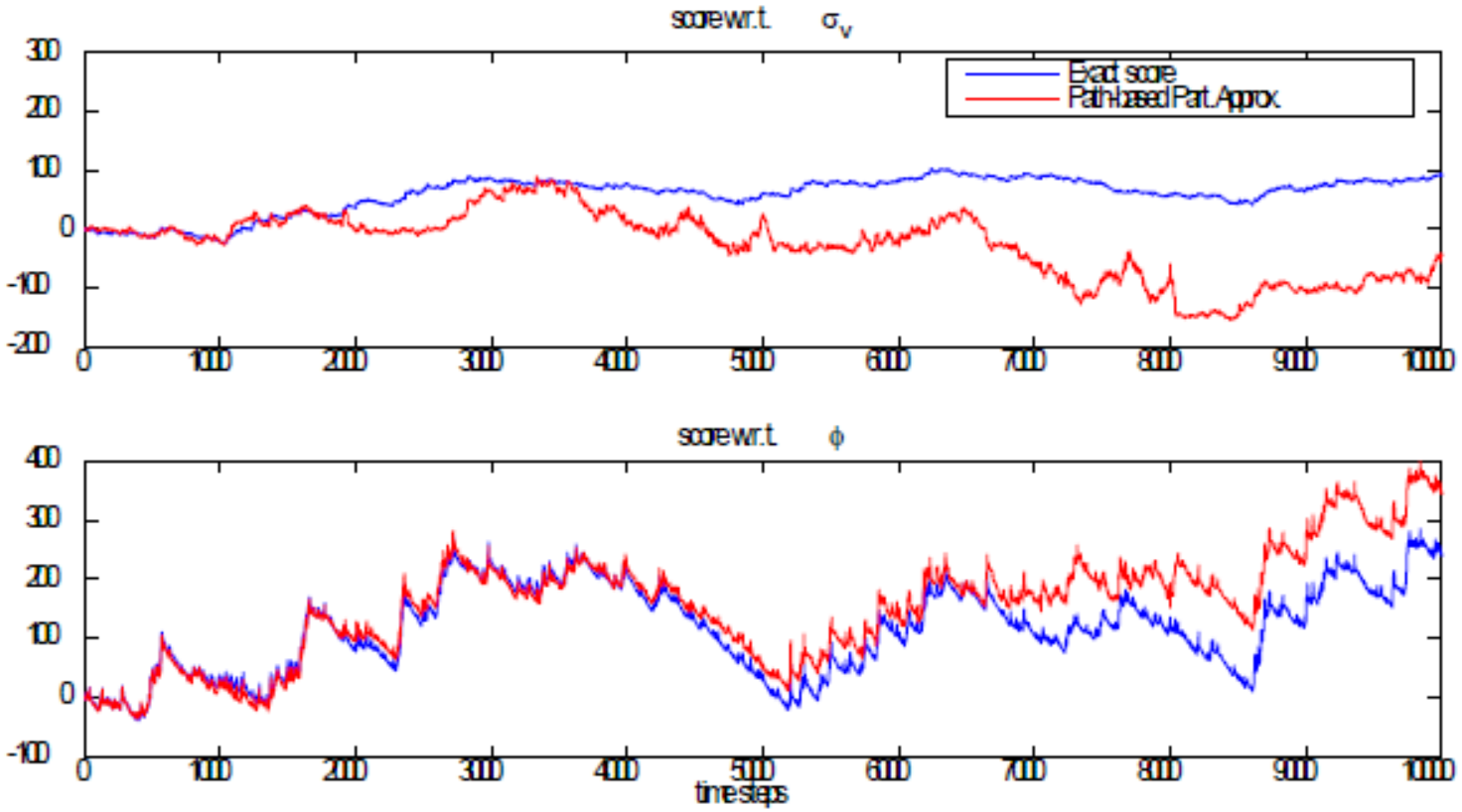
$$\alpha_n^{(i)} = \widehat{\nabla \log p_\theta}(\boldsymbol{X}_{1:n}^{(i)} \mid \boldsymbol{Y}_{1:n-1})$$

- This is simple but inefficient as based on importance sampling on spaces of increasing dimension and in addition it relies on being able to obtain a good approximation of $p_\theta(\boldsymbol{x}_{1:n} \mid \boldsymbol{Y}_{1:n-1})$

- F. Cérou, P. Del Moral, T. Furon, A. Guyader, Sequential MC for rare event estimation, Statistics and Computing, May 2012, Volume 22, Issue 3, pp 795–808

Degeneracy of the SMC Algorithm

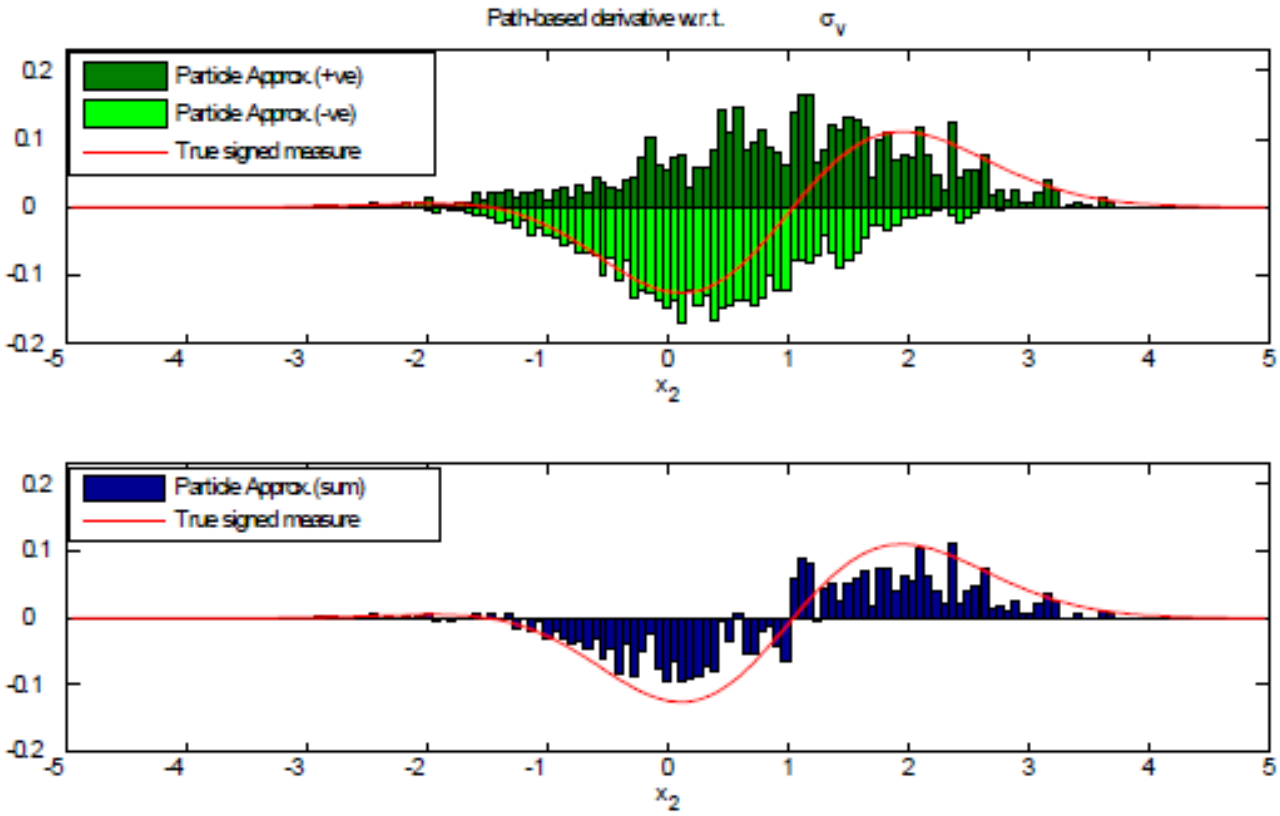
- Score $\nabla p_\theta(\boldsymbol{Y}_{1:n})$ for linear Gaussian models: exact (blue) vs SMC approximation (red).



$$X_n = \phi X_{n-1} + \sigma_v V_n$$
$$Y_n = X_n + \sigma_w W_n$$

Degeneracy of the SMC Algorithm

- Signed measure $\nabla_\theta p(x_n | \boldsymbol{Y}_{1:n})$ for linear Gaussian models: exact (red) vs SMC (blue/green). Positive and negative particles are mixed.



Marginal Importance Sampling for Sensitivity Estimation

- An alternative identity is the following:

$$\nabla p_\theta(x_n \mid \boldsymbol{Y}_{1:n-1}) = \nabla \log p_\theta(x_n \mid \boldsymbol{Y}_{1:n-1})\, p_\theta(x_n \mid \boldsymbol{Y}_{1:n-1})$$

- It suggests using an SMC approximation of the form:

$$\widehat{\nabla p_\theta}(x_n \mid \boldsymbol{Y}_{1:n-1}) = \frac{1}{N}\sum_{i=1}^{N} \beta_n^{(i)} \delta_{X_n^{(i)}}(x_n)$$

$$\beta_n^{(i)} = \widehat{\nabla \log p_\theta}(X_n^{(i)} \mid \boldsymbol{Y}_{1:n-1}) = \frac{\widehat{\nabla p_\theta}(X_n^{(i)} \mid \boldsymbol{Y}_{1:n-1})}{\hat{p}_\theta(X_n^{(i)} \mid \boldsymbol{Y}_{1:n-1})}$$

- Such an approximation relies on a pointwise approximation of both the filter and its derivative. The computational complexity is $\mathcal{O}(N^2)$ as

$$\hat{p}_\theta(X_n^{(i)} \mid \boldsymbol{Y}_{1:n-1}) \propto \int f_\theta(X_n^{(i)} \mid x_{n-1}) \hat{p}_\theta(x_{n-1} \mid \boldsymbol{Y}_{1:n-1}) dx_{n-1} = \frac{1}{N}\sum_{j=1}^{N} f_\theta(X_n^{(i)} \mid X_{n-1}^{(j)})$$

Marginal Importance Sampling for Sensitivity Estimation

- The optimal filter satisfies:

$$p_\theta(x_n \mid \boldsymbol{Y}_{1:n}) = \frac{\xi_\theta(x_n \mid \boldsymbol{Y}_{1:n})}{\int \xi_\theta(x_n \mid \boldsymbol{Y}_{1:n}) dx_n}$$

$$\xi_\theta(x_n \mid \boldsymbol{Y}_{1:n}) = g_\theta\left(Y_n \mid x_n\right) \int f_\theta\left(x_n \mid x_{n-1}\right) p_\theta(x_{n-1} \mid \boldsymbol{Y}_{1:n-1}) dx_{n-1}$$

- The derivatives satisfy the following:

$$\nabla p_\theta(x_n \mid \boldsymbol{Y}_{1:n}) = \frac{\nabla \xi_\theta(x_n \mid \boldsymbol{Y}_{1:n})}{\int \xi_\theta(x_n \mid \boldsymbol{Y}_{1:n}) dx_n} - p_\theta(x_n \mid \boldsymbol{Y}_{1:n}) \frac{\int \nabla \xi_\theta(x_n \mid \boldsymbol{Y}_{1:n}) dx_n}{\int \xi_\theta(x_n, \boldsymbol{Y}_{1:n}) dx_n}$$

- This way we obtain a simple recursion of $\nabla p_\theta(x_n \mid \boldsymbol{Y}_{1:n})$ as a function of $\nabla p_\theta(x_{n-1} \mid \boldsymbol{Y}_{1:n-1})$ and $p_\theta(x_{n-1} \mid \boldsymbol{Y}_{1:n-1})$.

SMC Approximation of the Sensitivity

- Sample

$$X_n^{(i)} \sim q_\theta(.|Y_n) = \sum_{j=1}^{N} \eta_n^{(j)} q_\theta\left(.|Y_n, X_{n-1}^{(j)}\right), \quad \eta_n^{(j)} \propto W_{n-1}^{(i)} \hat{p}\left(Y_n|X_{n-1}^{(i)}\right)$$

- Compute:

$$\alpha_n^{(i)} = \frac{\hat{\xi}_\theta(X_n^{(i)}, \boldsymbol{Y}_{1:n})}{q_\theta\left(X_n^{(i)}|Y_n\right)}, \quad \rho_n^{(i)} = \frac{\widehat{\nabla \xi}_\theta(X_n^{(i)}, \boldsymbol{Y}_{1:n})}{q_\theta\left(X_n^{(i)}|Y_n\right)}$$

$$W_n^{(i)} = \frac{\alpha_n^{(i)}}{\sum_{j=1}^{N} \alpha_n^{(j)}}, \quad W_n^{(i)} \beta_n^{(i)} = \frac{\rho_n^{(i)}}{\sum_{j=1}^{N} \alpha_n^{(j)}} - W_n^{(i)} \frac{\sum_{j=1}^{N} \rho_n^{(j)}}{\sum_{j=1}^{N} \alpha_n^{(j)}}$$

- We have:

$$\hat{p}_\theta(x_n|\boldsymbol{Y}_{1:n}) = \sum_{i=1}^{N} W_n^{(i)} \delta_{X_n^{(i)}}(x_n)$$

$$\widehat{\nabla p}_\theta(x_n|\boldsymbol{Y}_{1:n}) = \sum_{i=1}^{N} W_n^{(i)} \beta_n^{(i)} \delta_{X_n^{(i)}}(x_n)$$

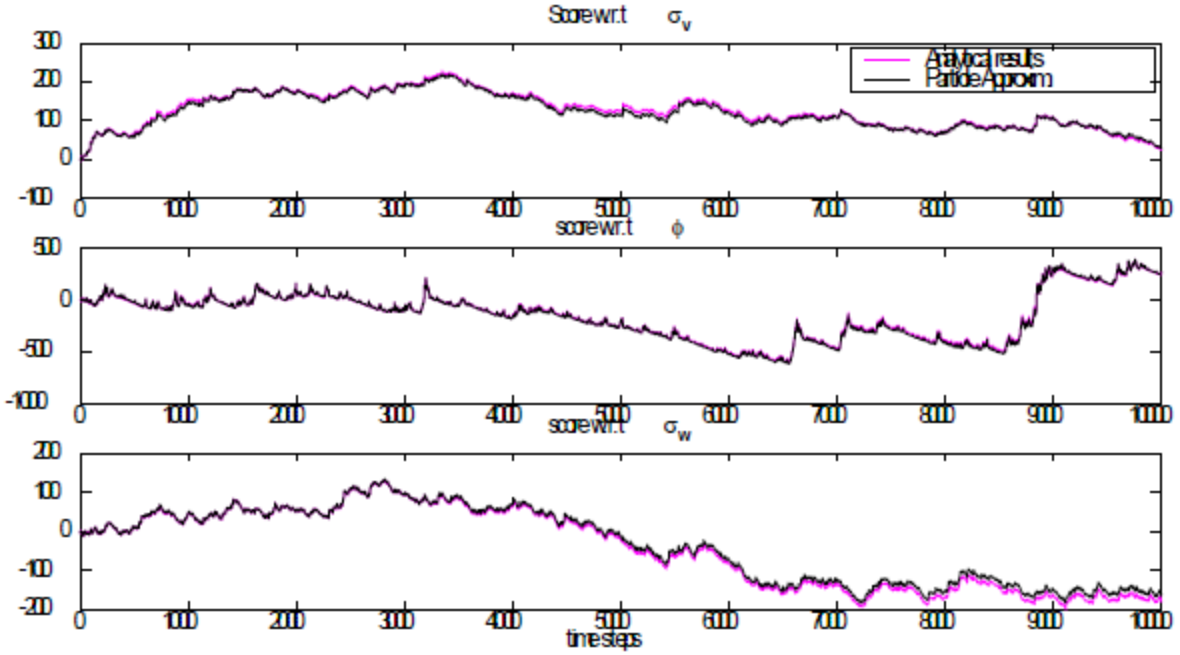
Example: Linear Gaussian Model

- Consider the following model:

$$X_n = \phi X_{n-1} + \sigma_v V_n$$
$$Y_n = X_n + \sigma_w W_n$$

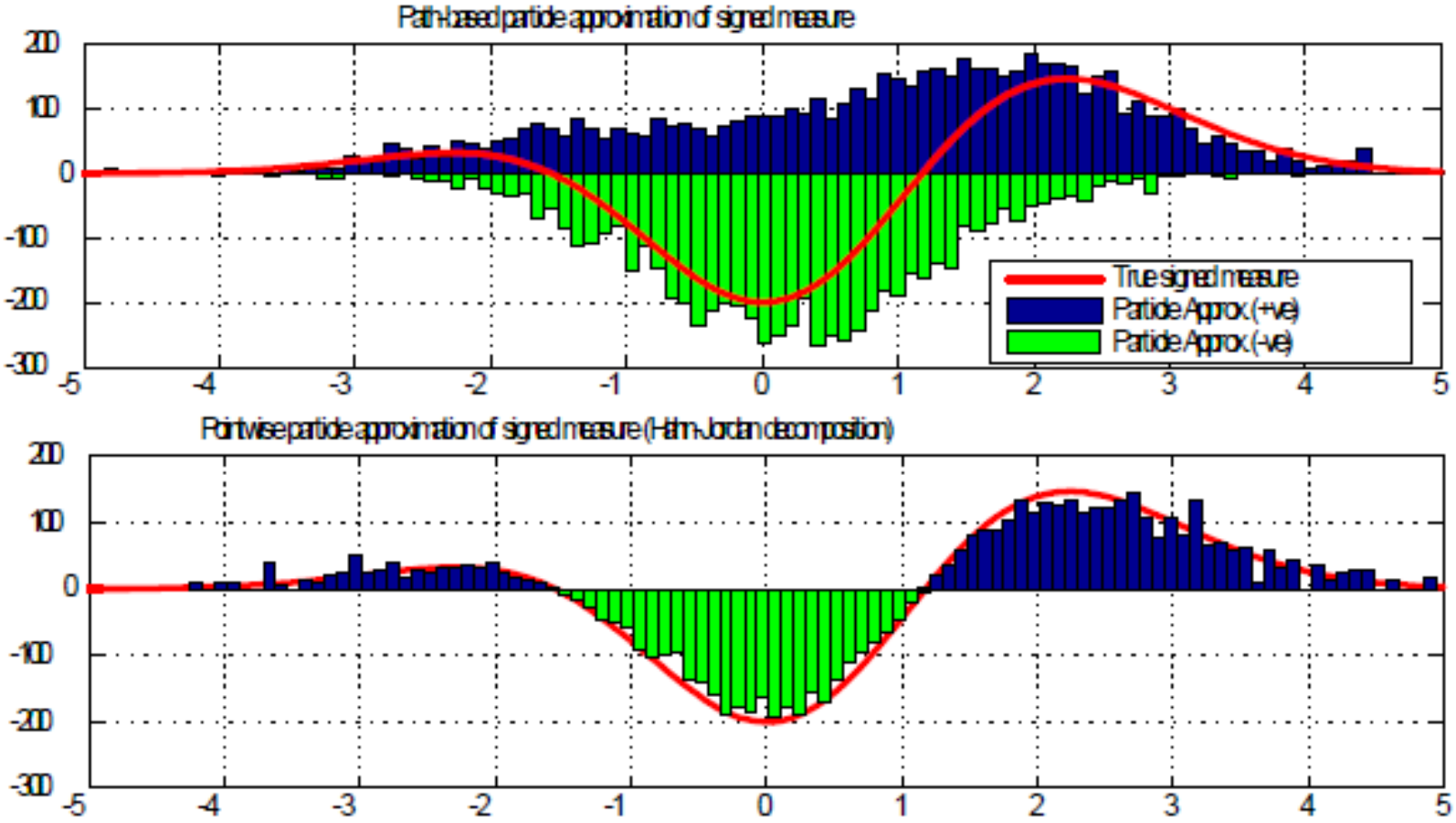
- We have $\theta = \{\phi, \sigma_v, \sigma_w\}$
- We compare the SMC particle approximation estimate $\widehat{\nabla \log p_\theta}(\boldsymbol{Y}_{1:n})$ for N=1000 to its exact value given by the Kalman filter derivative (exact with cyan vs SMC with black).

$\widehat{\nabla p_\theta}(\boldsymbol{Y}_{1:n})$



Example: Linear Gaussian Model

- Marginal of the signed measure (Hahn Jordan decomposition) of the joint (top) and Hahn Jordan of the marginal (bottom)



Example: Stochastic Volatility Model

- Consider the following model:

$$X_n = \phi X_{n-1} + \sigma_v V_n$$

$$Y_n = \beta \exp\left(\frac{X_n}{2}\right) W_n$$

- We have $\theta = \{\phi, \sigma_v, \beta\}$. We use SMC with N=1000 for batch and on-line estimation.

- For Batch Estimation:

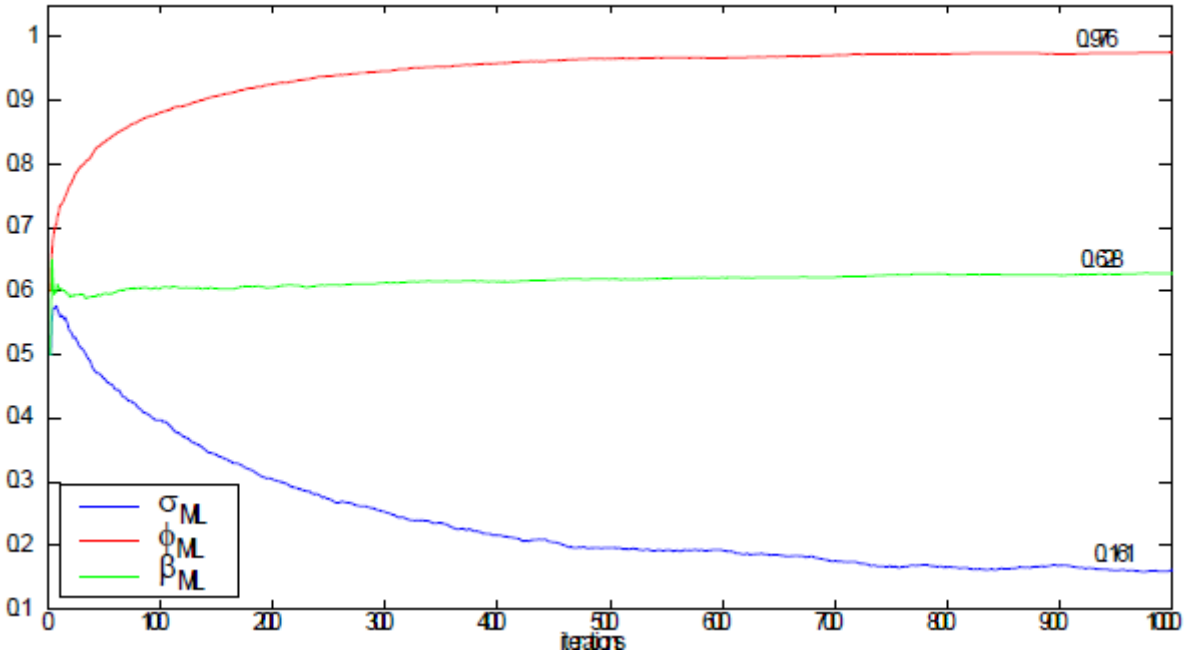
$$\theta_n = \theta_{n-1} + \gamma_n \widehat{\nabla \log p}_{\theta_{n-1}}(\boldsymbol{Y}_{1:n})$$

- For online estimation:

$$\theta_n = \theta_{n-1} + \gamma_n \widehat{\nabla \log p}_{\boldsymbol{\theta}_{1:n-1}}(Y_n | \boldsymbol{Y}_{1:n-1})$$

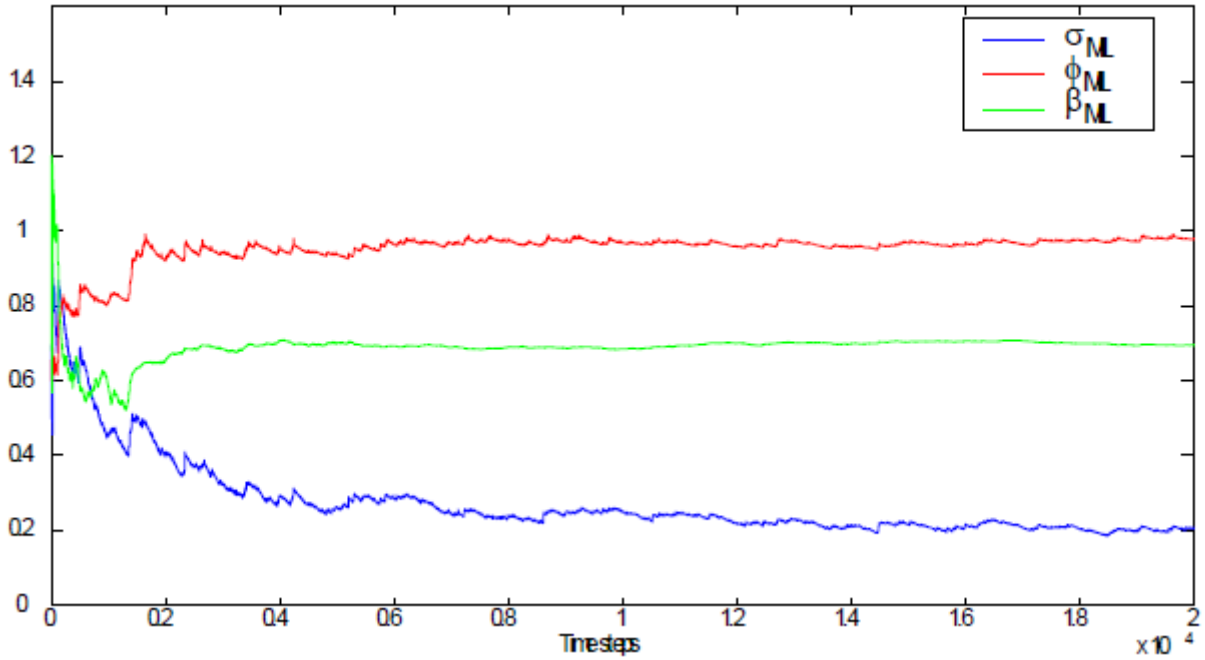
Example: Stochastic Volatility Model

- Batch Parameter Estimation for Daily Exchange Rate Pound/Dollar 81-85



Example: Stochastic Volatility Model

- Recursive parameter estimation for SV model. The estimates converge towards the true values.



Summary: Stochastic Gradient Algorithm

- The stochastic gradient algorithm is sensitive to initialization.
- The computational complexity is $\mathcal{O}(N^2)$ but can be further improved.
- Uniform convergence of the particle approximation has been established.
- It has been successfully applied to high-dimensional parameter estimation problems.

- Tadić, VB & Doucet, A 2005, 'Exponential forgetting and geometric ergodicity for optimal filtering in general state-space models' *Stochastic Processes and their Applications*, vol 115 (8), pp. 1408 - 1436. DOI: 10.1016/j.spa.2005.03.005

Sequential Monte Carlo Fixed-Lag Smoothing Approximation

- P. Del Moral, A. Doucet & S.S. Singh, Forward Smoothing using Sequential Monte Carlo, technical report, Cambridge University, 2009
- Godsill, Doucet, and West: Monte Carlo Smoothing for Nonlinear Time Series. 157, Journal of the American Statistical Association, March 2004, Vol. 99, No. 465.
- P. Del Moral, A. Doucet & S.S. Singh, Forward Smoothing using Sequential Monte Carlo, technical report, Cambridge University, 2009
- Paul Fearnhead David Wyncoll Jonathan Tawn *Biometrika*, Volume 97, Issue 2, 1 June 2010, Pages 447–464, A sequential smoothing algorithm with linear computational cost.
- Mark Briers Arnaud DoucetEmail author Simon Maskell, Smoothing algorithms for state–space models Annals of the Institute of Statistical Mathematics February 2010, 62:61

Smoothing Approximation

- Note that $p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) = p(x_n|\boldsymbol{y}_{1:n}) \prod_{k=1}^{n-1} p(x_k|\boldsymbol{x}_{k+1:n}, \boldsymbol{y}_{1:n})$. This defines the general smoothing problem.
  - Particular estimate of interest $p(x_k|\boldsymbol{y}_{1:n})$
- Godsill et al. proposed a solution to this smoothing problem by proposing a particle approximation to each term in the product $\prod_{k=1}^{n-1} p(x_k|\boldsymbol{x}_{k+1:n}, \boldsymbol{y}_{1:n})$.
- Using the Markov property note the following:

$$p(x_k|\boldsymbol{x}_{k+1:n}, \boldsymbol{y}_{1:n}) = p(x_k|x_{k+1}, \boldsymbol{y}_{1:k}) = \frac{p(x_k, x_{k+1}|\boldsymbol{y}_{1:k})}{p(x_{k+1}|\boldsymbol{y}_{1:k})}$$
$$\propto f(x_{k+1}|x_k)\, p(x_k|\boldsymbol{y}_{1:k})$$

- Here prior=$f(x_{k+1}|x_k)$ and filtering distribution = $p(x_k|\boldsymbol{y}_{1:k})$.

- Godsill, Doucet, and West: Monte Carlo Smoothing for Nonlinear Time Series. 157, Journal of the American Statistical Association, March 2004, Vol. 99, No. 465.

Smoothing Approximation

$$p(x_k|\boldsymbol{x}_{k+1:n}, \boldsymbol{y}_{1:n}) \propto f(x_{k+1}|x_k)p(x_k|\boldsymbol{y}_{1:k})$$

- Here prior=$f(x_{k+1}|x_k)$ and filtering distribution = $p(x_k|\boldsymbol{y}_{1:k})$.

- Given a particle approximation $\left\{W_k^{(i)}, x_k^{(i)}\right\}_{i=1}^N$ of the filtering density $p(x_k|\boldsymbol{y}_{1:k})$ and a sample X_{k+1}, we can obtain a particle approximation of $p(x_k|\boldsymbol{x}_{k+1:n}, \boldsymbol{y}_{1:n})$ by updating the weights as follows:

$$w_{k|k+1}^{(i)} = W_k^{(i)} f\left(X_{k+1}|X_k^{(i)}\right) \text{ and } W_{k|k+1}^{(i)} = \frac{w_{k|k+1}^{(i)}}{\sum_i w_{k|k+1}^{(i)}}$$

- Recalling that $p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) = p(x_n|\boldsymbol{y}_{1:n}) \prod_{k=1}^{n-1} p(x_k|\boldsymbol{x}_{k+1:n}, \boldsymbol{y}_{1:n})$, we can implement an algorithm for computing a particle approximation of $p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$.

Smoothing Approximation: Algorithm

$$p(x_k|\boldsymbol{x}_{k+1:n}, \boldsymbol{y}_{1:n}) \propto f(x_{k+1}|x_k)p(x_k|\boldsymbol{y}_{1:k})$$

$$w_{k|k+1}^{(i)} = W_k^{(i)} f\left(X_{k+1}|X_k^{(i)}\right) \text{ and } W_{k|k+1}^{(i)} = \frac{w_{k|k+1}^{(i)}}{\sum_i w_{k|k+1}^{(i)}}$$

$$p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) = p(x_n|\boldsymbol{y}_{1:n}) \prod_{k=1}^{n-1} p(x_k|\boldsymbol{x}_{k+1:n}, \boldsymbol{y}_{1:n})$$

- Draw $\hat{X}_n$ from the particle approximation $\left\{W_n^{(i)}, x_n^{(i)}\right\}_{i=1}^N$ of $p(x_n|\boldsymbol{y}_{1:n})$
- For $k = n-1, \ldots, 1$
  - Calculate $w_{k|k+1}^{(i)} = W_k^{(i)} f\left(\hat{X}_n|X_k^{(i)}\right)$ for i=1,2,…,n
  - Draw $\hat{X}_k$ with probability $\propto w_{k|k+1}^{(i)}$
- Then $\hat{\boldsymbol{X}}_{1:n} = \{\hat{X}_1, \ldots, \hat{X}_n\}$ is a sample from $p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$
- Further independent realizations are obtained by repeating this procedure as many times as needed.

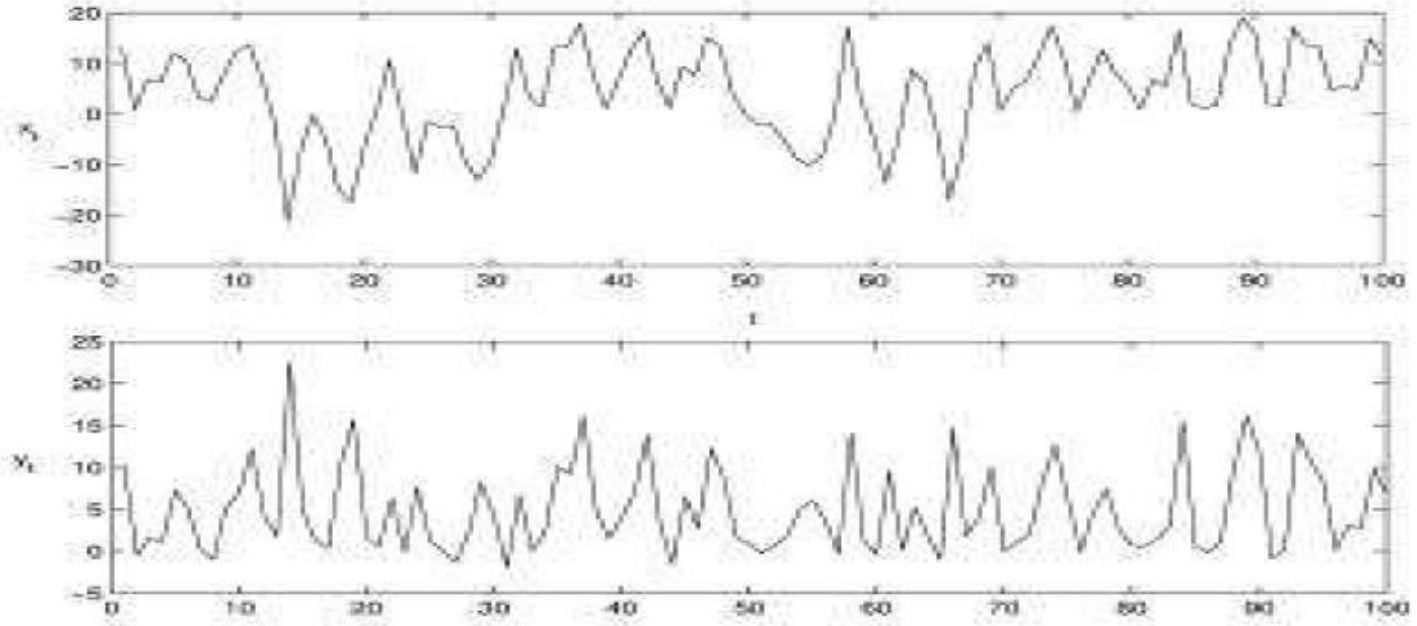
Smoothing Approximation

$$X_k = \frac{X_{k-1}}{2} + 25\frac{X_{k-1}}{1+X_{k-1}^2} + 8cos(1.2k) + u_k,\ u_k \sim \mathcal{N}(0,10)$$

$$Y_k = \frac{X_k^2}{20} + w_k,\ w_k \sim \mathcal{N}(0,1)$$

$x_n\ (true)$

y_n



- Godsill, Doucet, and West: Monte Carlo Smoothing for Nonlinear Time Series. 157, Journal of the American Statistical Association, March 2004, Vol. 99, No. 465.

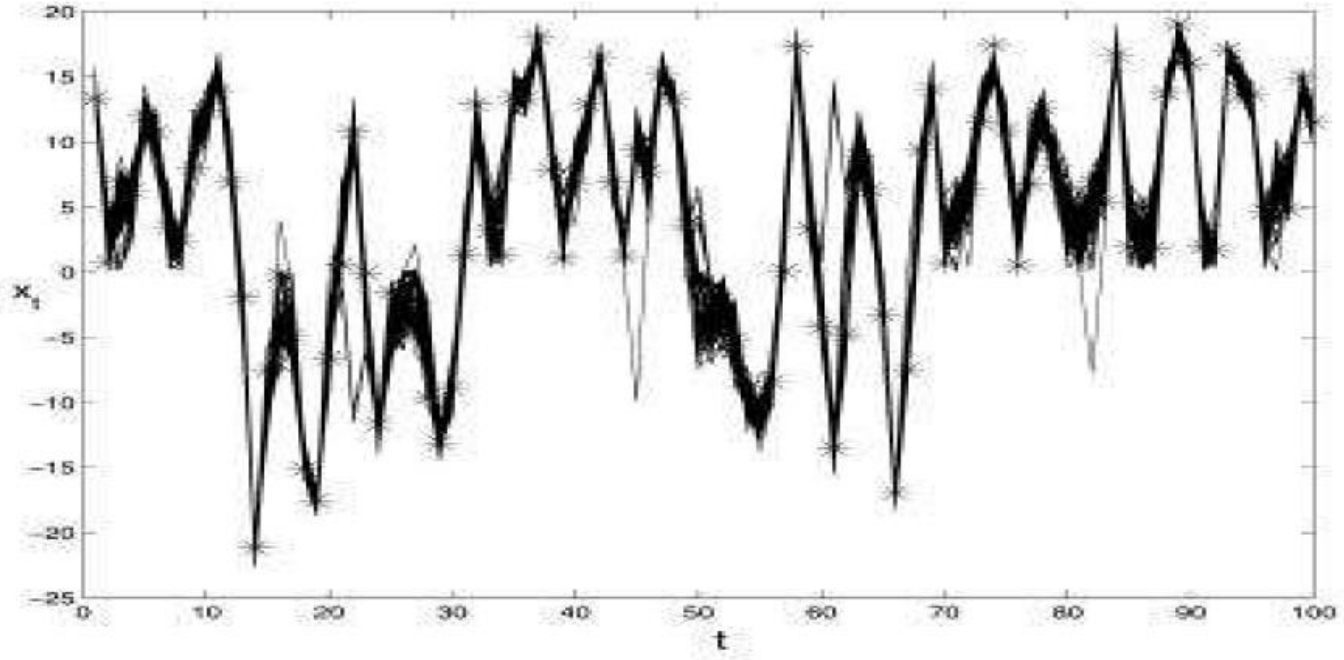
Smoothing Approximation

$$X_k = \frac{X_{k-1}}{2} + 25\frac{X_{k-1}}{1+X_{k-1}^2} + 8cos(1.2k) + u_k,\ u_k \sim \mathcal{N}(0,10)$$

$$Y_k = \frac{X_k^2}{20} + w_k,\ w_k \sim \mathcal{N}(0,1)$$

Smoothing trajectories drawn from

$p(\boldsymbol{x}_{1:100}|\boldsymbol{y}_{1:100})$



- Godsill, Doucet, and West: Monte Carlo Smoothing for Nonlinear Time Series. 157, Journal of the American Statistical Association, March 2004, Vol. 99, No. 465.

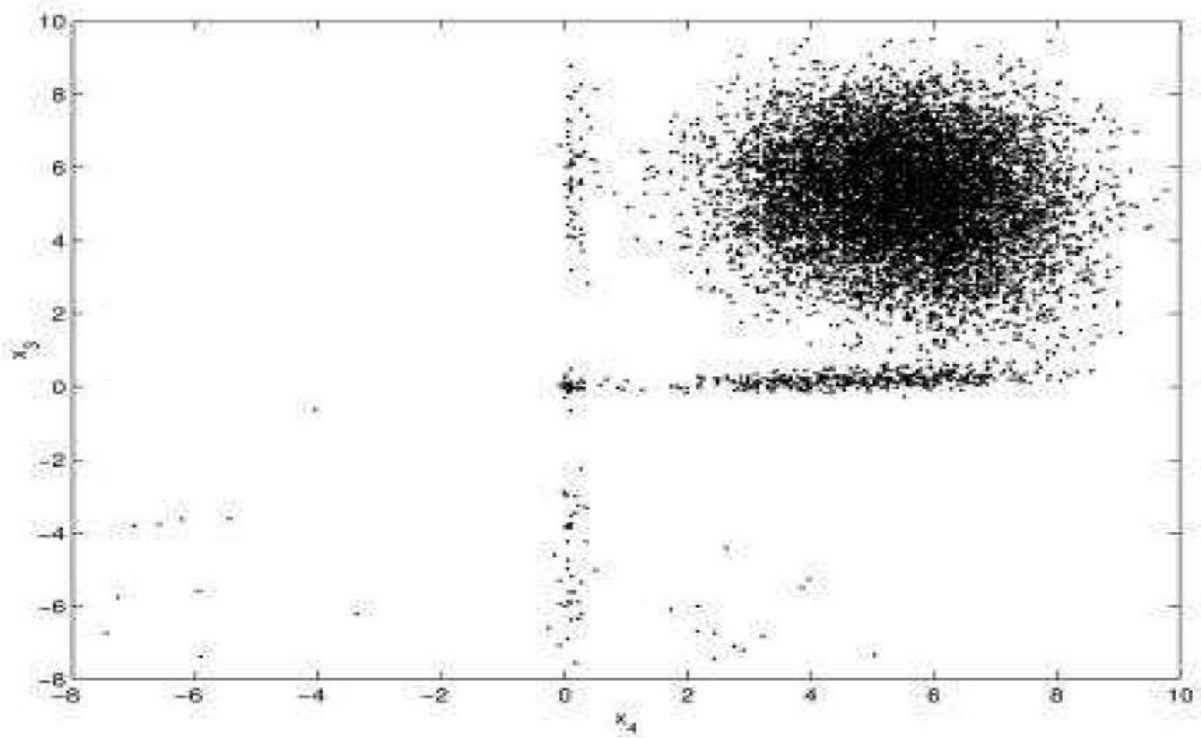
Smoothing Approximation

$$X_k = \frac{X_{k-1}}{2} + 25\frac{X_{k-1}}{1+X_{k-1}^2} + 8cos(1.2k) + u_k,\ u_k \sim \mathcal{N}(0,10)$$

$$Y_k = \frac{X_k^2}{20} + w_k,\ w_k \sim \mathcal{N}(0,1)$$

Scatter plot of samples drawn from

$p(\boldsymbol{x}_{3:4}|\boldsymbol{y}_{1:100})$



- Godsill, Doucet, and West: Monte Carlo Smoothing for Nonlinear Time Series. 157, Journal of the American Statistical Association, March 2004, Vol. 99, No. 465.

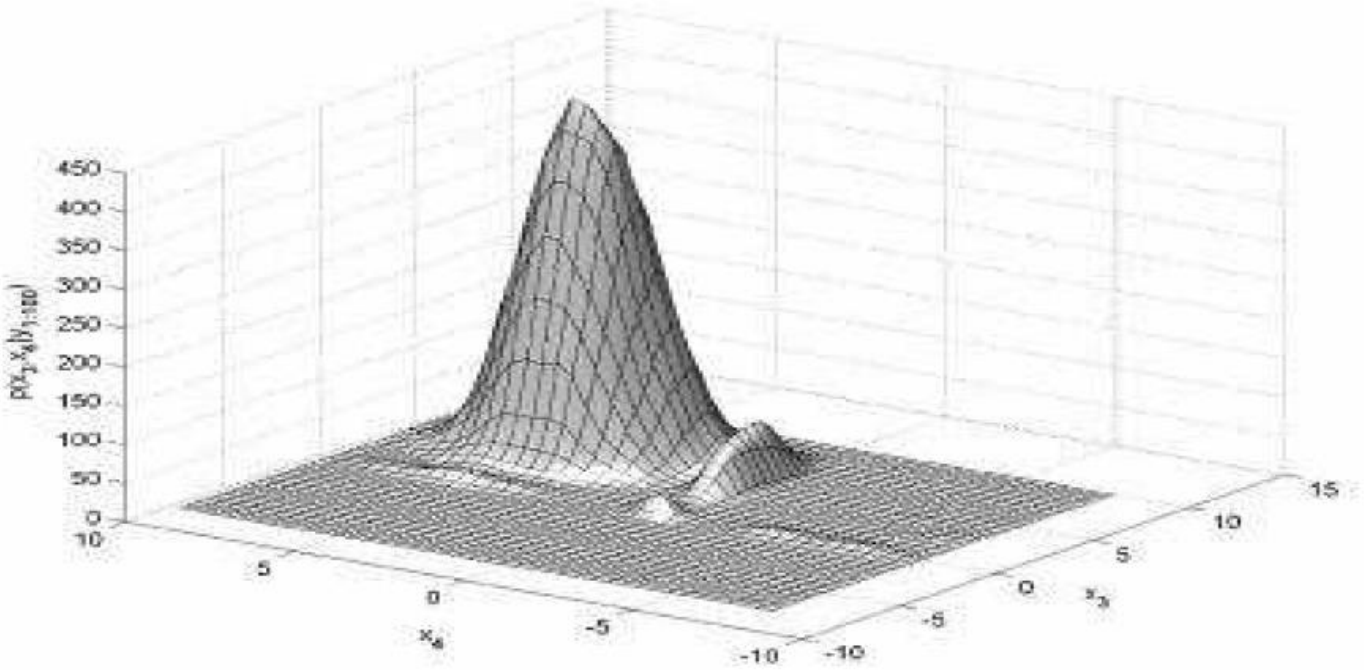
Smoothing Approximation

$$X_k = \frac{X_{k-1}}{2} + 25\frac{X_{k-1}}{1+X_{k-1}^2} + 8cos(1.2k) + u_k,\ u_k \sim \mathcal{N}(0,10)$$

$$Y_k = \frac{X_k^2}{20} + w_k,\ w_k \sim \mathcal{N}(0,1)$$

Kernel density estimate of

$p(\boldsymbol{x}_{3:4}|\boldsymbol{y}_{1:100})$



- Godsill, Doucet, and West: Monte Carlo Smoothing for Nonlinear Time Series. 157, Journal of the American Statistical Association, March 2004, Vol. 99, No. 465.

Fixed-Lag Smoothing Approximation

- Direct SMC approximation of $p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$ and its marginal $p(x_k|\boldsymbol{y}_{1:n})$ gets worst as n increases.

- The fixed-lag smoothing approximation relies on the following:

$$p(\boldsymbol{x}_{1:k}|\boldsymbol{y}_{1:n}) \approx p(\boldsymbol{x}_{1:k}|\boldsymbol{y}_{1:k+\Delta}), \text{ for } \Delta \text{ large}$$

- In this case we stop resampling $\left\{X_k^{(i)}\right\}$ after time $k+\Delta$

- Computational cost $\mathcal{O}(Nn)$ but non-vanishing bias as $N \to \infty$.

- The selection of Δ is difficult.

  - Too small Δ results in $p(\boldsymbol{x}_{1:k}|\boldsymbol{y}_{1:k+\Delta})$ being a poor approximation of $p(\boldsymbol{x}_{1:k}|\boldsymbol{y}_{1:n})$

  - Too large Δ improves the approximation but brings in degeneracy.

SMC Forward Filtering Backward Smoothing

$$\underbrace{p\left(x_k \mid \boldsymbol{y}_{1:n}\right)}_{\text{Smoother at } k} = \int \underbrace{p\left(x_{k+1} \mid \boldsymbol{y}_{1:n}\right)}_{\text{Smoother at } k+1} \underbrace{\frac{f\left(x_{k+1} \mid x_k\right) \overbrace{p\left(x_k \mid \boldsymbol{y}_{1:k}\right)}^{\text{Filter at } k}}{p\left(x_{k+1} \mid \boldsymbol{y}_{1:n}\right)}}_{\text{backward transition } p\left(x_k \mid \boldsymbol{y}_{1:n}, x_{k+1}\right)} dx_{k+1}$$

- For $k = 1, \ldots, n$, compute $\hat{p}(x_k|\boldsymbol{y}_{1:k})$
- For $k = n-1, \ldots, 1$, compute $\hat{p}(x_k|\boldsymbol{y}_{1:n}) = \sum_{j=1}^{N} W_{k|n}^{(i)} \delta_{X_k^{(i)}}(x_k)$ with cost $\mathcal{O}(N^2 n)$ using:

$$W_{k|n}^{(i)} = \sum_{j=1}^{N} W_{k+1|n}^{(i)} \frac{f\left(X_{k+1}^{(j)} \mid X_k^{(i)}\right)}{\sum_{l=1}^{N} f\left(X_{k+1}^{(j)} \mid X_k^{(l)}\right)}$$

- For $\phi_n\left(\boldsymbol{x}_{1:n}\right) = \sum_{k=1}^{n-1} s_k\left(x_k, x_{k+1}\right)$, the SMC FFBS estimates $\{\hat{\phi}_n\}$ are exact.
- Sampling from $\hat{p}(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$ costs $\mathcal{O}(Nn)$ but $\mathcal{O}(n)$ through rejection sampling.

SMC Generalized Two-Filter Smoothing

$$p\left(x_k, x_{k+1} \mid \boldsymbol{y}_{1:n}\right) \propto \frac{\overbrace{p\left(x_k \mid \boldsymbol{y}_{1:n}\right)}^{\text{Forward filter}} f\left(x_{k+1} \mid x_k\right) \overbrace{\overline{p}\left(x_{k+1} \mid \boldsymbol{y}_{k+1:n}\right)}^{\text{Generalized Backward filter}}}{\underbrace{\overline{p}\left(x_{k+1}\right)}_{\text{Artificial Prior}}}$$

- For $k = 1, \ldots, n$, compute : $\hat{p}(x_k|\boldsymbol{y}_{1:k})$
- For $k = n, \ldots, 1$, compute $\hat{\overline{p}}(x_{k+1}|\boldsymbol{y}_{\mathrm{k+1:n}})$. Combine the forward and backward filters to obtain:

$$\hat{p}(x_k, x_{k+1}|\boldsymbol{y}_{1:n}) \propto \hat{p}(x_k|\boldsymbol{y}_{1:k}) \frac{f(x_{k+1}|x_k)}{\bar{p}(x_{k+1})} \hat{\overline{p}}(x_{k+1}|\boldsymbol{y}_{k+1:n})$$

- Cost $\mathcal{O}(N^2n)$ but $\mathcal{O}(n)$ through rejection sampling (Briers et al.) and importance sampling (Fearnhead et al.).

- Paul Fearnhead David Wyncoll Jonathan Tawn *Biometrika*, Volume 97, Issue 2, 1 June 2010, Pages 447–464, A sequential smoothing algorithm with linear computational cost.
- Mark Briers Arnaud Doucet, Simon Maskell, Smoothing algorithms for state–space models Annals of the Institute of Statistical Mathematics February 2010, 62:61