Introduction to Sequential Monte Carlo Methods

Importance Sampling for Nonlinear Non-Gaussian Dynamic Models

Prof. Nicholas Zabaras
Center for Informatics and Computational Science
https://cics.nd.edu/
University of Notre Dame
Notre Dame, Indiana, USA

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

November 6, 2018

Contents

- References and SMC Resources
- INTRODUCING THE STATE SPACE MODEL: Discrete time Markov Models, Tracking Problem, Speech Enhancement, the Dynamics of an Asset, The state space model with observations, target distribution and point estimates, particle motion in random field
- BAYESIAN INFERENCE IN STATE SPACE MODELS: Bayesian recursion for the state-space model, Forward Filtering, Forward-Backward Filter
- ONLINE BAYESIAN PARAMETER ESTIMATION: Online parameter estimation, MLE, EM, Need for numerical approach to HMM
- IMPORTANCE SAMPLING: Review of IS, Monte Carlo for the State Space Example, IS for the State Space Model, Optimal Importance Distribution, Effective Sample Size, Sequential IS

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
- 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.

Introduction

- Sequential Monte Carlo (SMC) methods are used to approximate any sequence of probability distributions.

- They are used often in physics

  - Compute eigenvalues of positive operators
  - Compute free energies
  - Solve differential or integral equations
  - Simulate polymer chains
  - Etc.

- Hidden Markov Models (HMM) are used in these notes and most tutorials for introducing SMC – but SMC is clearly a method for a much bigger class of problems.

- In HMM, SMC methods are often known as Particle Filtering or Smoothing Methods.

Introducing the State Space Model

Discrete-Time Markov Model

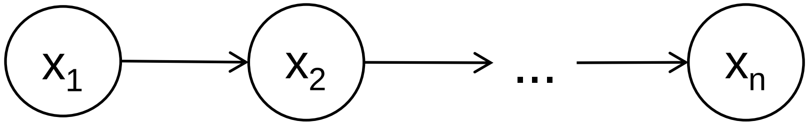
- ❑ Consider a discrete-time Markov process : $\{X_n\}, n \geq 1$

- ❑ It is defined by an initial density $X_1 \sim \mu(.)$ and a transition density:

$$X_n \mid \left(X_{n-1} = x\right) \sim f\left(\cdot \mid x\right)$$

- ❑ We then can write (prior distribution of the states):

$$p\left(\boldsymbol{x}_{1:n}\right) \equiv p\left(x_1, \cdots, x_n\right) = \mu(x_1) \prod_{k=2}^{n} f\left(x_k \mid x_{k-1}\right)$$



Markov Chain

Tracking Example

- ❑ Consider tracking a target in the XY plane (location/speed in $x-y$):

$$X_k = \left(X_{k,1}, V_{k,1}, X_{k,2}, V_{k,2}\right)^T$$

- ❑ We consider the constant velocity model:

$$\boldsymbol{X}_k = \boldsymbol{A}\boldsymbol{X}_{k-1} + \boldsymbol{W}_k, \boldsymbol{W}_k \overset{i.i.d.}{\sim} \mathcal{N}(0, \boldsymbol{\Sigma})$$

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_{CV} & 0 \\ 0 & \boldsymbol{A}_{CV} \end{pmatrix}, \boldsymbol{A}_{CV} = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix}$$

$$\boldsymbol{\Sigma} = \sigma^2 \begin{pmatrix} \boldsymbol{\Sigma}_{CV} & 0 \\ 0 & \boldsymbol{\Sigma}_{CV} \end{pmatrix}, \boldsymbol{\Sigma}_{CV} = \begin{pmatrix} T^3/3 & T^2/2 \\ T^2/2 & T \end{pmatrix}$$

- ❑ The transition density for this model is then:

$$f\left(x_k \mid x_{k-1}\right) = \mathcal{N}\left(x_k; \boldsymbol{A}x_{k-1}, \boldsymbol{\Sigma}\right)$$

Speech Enhancement

- ❑ We model speech signals as an autoregressive (AR) process, i.e.

$$S_k = \sum_{i=1}^{d} \alpha_i S_{k-i} + V_k, V_k \sim \mathcal{N}(0, \sigma_s^2)$$

- ❑ We can write this in a matrix form as follows:

$$U_k = \boldsymbol{A} U_{k-1} + \boldsymbol{B} V_k, U_k = \left(S_k, ..., S_{k-d}\right)^T$$

$$\boldsymbol{A} = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_d \\ 1 & & & \\ & \dots & & \\ & & 1 & \end{pmatrix}, \boldsymbol{B} = \begin{pmatrix} 1 \\ 0 \\ : \\ 0 \end{pmatrix}$$

- ❑ The transition density is now:

$$f_U\left(u_k \mid u_{k-1}\right) = \mathcal{N}\left(\left(u_k\right)_1; \left(\boldsymbol{A} u_{k-1}\right)_1, \sigma_s^2\right) \delta_{\left(u_{k-1}\right)_{1:d-1}}\left(\left(u_k\right)_{2:d}\right)$$

Speech Enhancement

- We can also consider the AR coefficients to be time dependent:

$$\alpha_k = \alpha_{k-1} + W_k, W_k \sim \mathcal{N}(0, \sigma_\alpha^2 I_d), \textit{where}:$$

$$\alpha_k = \left(\alpha_{k,1}, \ldots, \alpha_{k,d}\right)^T$$

- Thus for non-stationary speech signals, we can write:

$$f_\alpha\left(\alpha_k \mid \alpha_{k-1}\right) = \mathcal{N}(\alpha_k; \alpha_{k-1}, \sigma_\alpha^2 \boldsymbol{I}_d)$$

- The process $X_k = (a_k, Uk)$ is a Markov with transition density

$$f\left(x_k \mid x_{k-1}\right) = \mathcal{N}(\alpha_k; \alpha_{k-1}, \sigma_\alpha^2 I_d)\mathcal{N}\left(\left(u_k\right)_1; \left(\mathbf{A}_k u_{k-1}\right)_1, \sigma_s^2\right)\delta_{\left(u_{k-1}\right)_{1:d-1}}\left(\left(u_k\right)_{2:d}\right)$$

with

$$\left(A_k u_{k-1}\right)_1 = \left(\alpha_{k,1}, \ldots, \alpha_{k,d}\right)^T \begin{pmatrix} S_{k-1} \\ : \\ S_{k-1-d} \end{pmatrix}$$

Econometrics

- The Heston model (1993) describes the dynamics of an asset price S_t with the following model for $X_t = \log(St)$

$$dX_t = \mu dt + dW_t + dZ_t$$

where Z_t is a jump process, and dW_t Brownian motion.

- We approximate this (time integration) by a discrete-time Markov process

$$X_{t+\delta} = X_t + \delta\mu + \mathrm{W}_{t+\delta,\mathrm{t}} + \mathrm{Z}_{t+\delta,\mathrm{t}}$$

- The same model is used for biochemical networks, disease and population dynamics, etc.

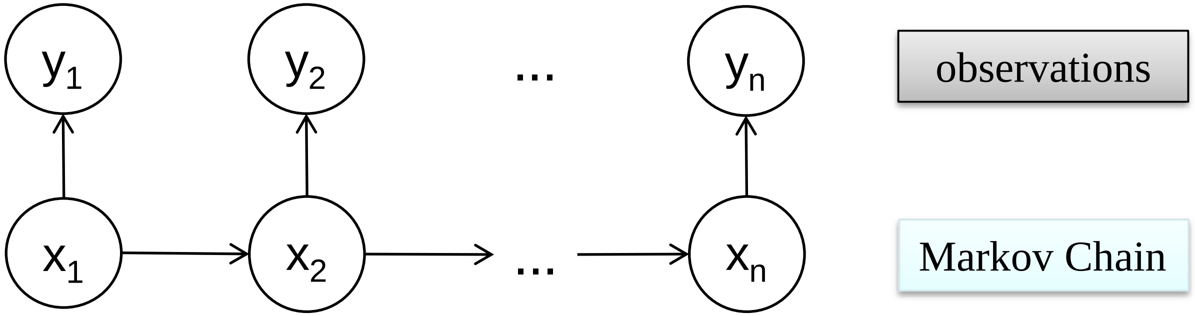
- D. Wilkison, Stochastic Modelling for Systems Biology, Second Edition, 2006
- E. Ionides, Inference for Nonlinear Dynamical Systems, PNAS, 2006

The State Space Model

- ❑ Let us discuss in some detail a very popular dynamic system, *the state space model,* now including observations

- ❑ A state space model is an extension of a Markov Chain which is able to capture the sequential relations among hidden variables.

- ❑ It is a dynamic system including two major parts



The State Space Model

- The two parts can be expressed by equations
  - state equation: $\{X_n\}, n \geq 1$ is a latent/hidden Markov process with

$$X_1 \sim \mu(.) \text{ and } X_n \mid \left(X_{n-1} = x_{n-1}\right) \sim f\left(\cdot \mid x_{n-1}\right)$$

  - observation equation: $\{Y_n\}, n \geq 1$ is an observation process with the observations being conditionally independent given $\{X_n\}, n \geq 1$

$$Y_n \mid \left(X_n = x_n\right) \sim g\left(\cdot \mid x_n\right)$$

- The observations $\{y_n\}$ are conditionally independent given the Markov states $\{x_n\}$, e.g. $g\left(y_i \mid x_i\right)$ and $g\left(y_j \mid x_j\right)$ are independent. Thus the likelihood is

$$p\left(y_1, \cdots, y_n \mid x_1, \cdots, x_n\right) = \prod_{i=1}^{n} g\left(y_i \mid x_i\right)$$

- Our aim is to recover $\{X_n\}, n \geq 1$ *given* $\{Y_n\}, n \geq 1$.

The State Space Model: Examples

- ❑ A Linear Gaussian State Space Model

$$X_1 \sim \mathcal{N}(m_1, \Sigma_1) \text{ and } X_n = AX_{n-1} + BV_n$$

$$Y_n = CX_n + DW_n, \text{ where}$$

$$V_n \overset{i.i.d.}{\sim} \mathcal{N}(0, \Sigma_v) \text{ and } W_n \overset{i.i.d.}{\sim} \mathcal{N}(0, \Sigma_w)$$

- ❑ A Stochastic Volatility Model

$$X_1 \sim \mathcal{N}\left(0, \frac{\sigma^2}{1-\alpha^2}\right) \text{and } X_n = \alpha X_{n-1} + V_n$$

$$Y_n = \beta \exp\left(X_n / 2\right) W_n, \text{ where}$$

$$|\alpha| < 1, V_n \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2) \text{ and } W_n \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$$

$$x_n \sim \mathcal{N}\left(\alpha x_{n-1}, \sigma^2\right) \quad g\left(y_n \mid x_n\right) = \mathcal{N}\left(y_n; 0, \beta^2 \exp\left(x_n\right)\right)$$

Tracking Example

- The simplest linear model is of the form:

$$Y_k = CX_k + E_k,\ E_k \overset{i.i.d.}{\sim} \mathcal{N}(0,\Sigma_e) \Rightarrow$$
$$g(y_k \mid x_k) = \mathcal{N}(y_k; CX_k, \Sigma_e)$$

- The non-linear version (Bearings-only-tracking) is more popular:

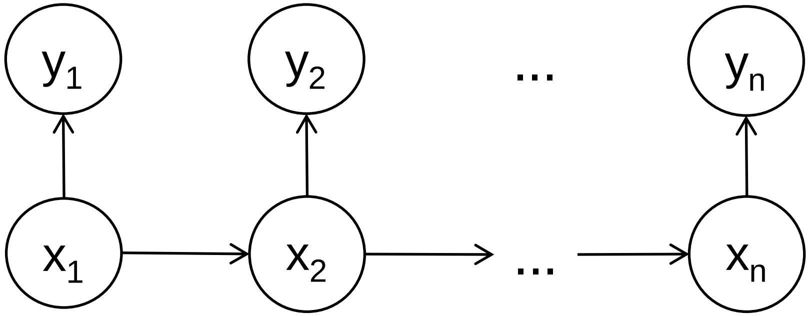
$$Y_k = \tan^{-1}\frac{X_{k,2}}{X_{k,1}} + E_k,\ E_k \overset{i.i.d.}{\sim} \mathcal{N}(0,\sigma^2) \Rightarrow$$
$$g(y_k \mid x_k) = \mathcal{N}\left(y_k; \tan^{-1}\frac{x_{k,2}}{x_{k,1}}, \sigma^2\right)$$

- Note that the mean of the Gaussian is a highly non-linear function of the state.

The State Space Model

- ❑ At time n, we have a total of n observations and the target distribution to be estimated is $p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$

- ❑ The target distribution is "time-varying". The posterior distribution should be updated after new observations are added. Thus we need to estimate a sequence of distributions according to the time sequence



observations

Markov Chain

Target Distribution

$p(x_1 \mid y_1)$ $\quad$ $p(\boldsymbol{x}_{1:2} \mid \boldsymbol{y}_{1:2})$ $\quad$ $p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$

$$Likelihood: p(y_1, \cdots, y_n \mid x_1, \cdots, x_n) = \prod_{i=1}^{n} g(y_i \mid x_i)$$

$$Prior: p(\boldsymbol{x}_{1:n}) = \mu(x_1) \prod_{k=2}^{n} f(x_k \mid x_{k-1})$$

Bayesian Inference in State Space Models

Bayesian Inference in State-Space Models

- In Bayesian estimation, the target distribution (posterior) for such state-space model is $p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)$

- The state equation for the Markov process defines a prior as

$$p\left(\boldsymbol{x}_{1:n}\right)=\mu\left(x_1\right)\prod_{k=2}^{n} f\left(x_k \mid x_{k-1}\right)$$

- The observation equation defines the likelihood as

$$p\left(\boldsymbol{y}_{1:n} \mid \boldsymbol{x}_{1:n}\right)=\prod_{k=1}^{n} g\left(y_k \mid x_k\right)$$

- The posterior distribution is known up to a normalizing constant

$$p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)=\frac{p\left(\boldsymbol{x}_{1:n},\boldsymbol{y}_{1:n}\right)}{p\left(\boldsymbol{y}_{1:n}\right)} \propto p\left(\boldsymbol{x}_{1:n},\boldsymbol{y}_{1:n}\right)=\underbrace{p\left(\boldsymbol{x}_{1:n}\right)}_{\Pr ior}\underbrace{p\left(\boldsymbol{y}_{1:n} \mid \boldsymbol{x}_{1:n}\right)}_{Likelihood}=\mu\left(x_1\right)\prod_{k=2}^{n} f\left(x_k \mid x_{k-1}\right)\prod_{k=1}^{n} g\left(y_k \mid x_k\right) \; and$$

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

Bayesian Inference in State-Space Models

- In this lecture, our target distribution is as follows:

$$\pi_n\left(\mathbf{x}_{1:n}\right)=\frac{\gamma_n\left(\mathbf{x}_{1:n}\right)}{Z_n}=p\left(\mathbf{x}_{1:n}\,|\,\mathbf{y}_{1:n}\right),\gamma_n\left(\mathbf{x}_{1:n}\right)=p\left(\mathbf{x}_{1:n},\mathbf{y}_{1:n}\right),\,Z_n=p\left(\mathbf{y}_{1:n}\right)$$

- The posterior and marginal likelihood do not admit close forms unless $\{X_n\}$ and $\{Y_n\}$ follow linear Gaussian equations or when $\{X_n\}$ takes values in a finite state space.

Bayesian Inference in State-Space Models

- From the posterior distribution, one can compute useful point estimates

$$\arg\max \, p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)$$

- One can also compute the MAP estimate for components

$$\arg\max \, p\left(x_k \mid \boldsymbol{y}_{1:n}\right)$$

$$p\left(x_k \mid \boldsymbol{y}_{1:n}\right) = \int \ldots \int p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right) d\boldsymbol{x}_{1:k-1} d\boldsymbol{x}_{k+1:n}$$

- The posterior mean (minimum mean square estimate) can also be estimated as:

$$\mathbb{E}\left[X_k \mid \boldsymbol{y}_{1:n}\right] = \int x_k \, p\left(x_k \mid \boldsymbol{y}_{1:n}\right) dx_k$$

Particle Motion in Random Medium

- Consider a Markovian particle $\{X_n\}, n \geq 1$ evolving in a random medium as follows:

$$X_1 \sim \mu(.)\ and\ X_{n+1} | \left(X_n = x\right) \sim f\left(\cdot | x\right)$$

- At time n, the probability for the particle to be killed is given as $1 - g(X_n)$, where $0 \leq g(x) \leq 1$ for any $x \in E$.

- Let T be the time at the which the particle is killed. We want to compute the probability $\Pr(T > n)$.

Particle Motion in Random Medium

- Starting from $t = 1$, given the current state x_1, the probability for the particle to survive is given as $g(x_1)$.

- Thus, the joint probability {particle at state x_1, particle survive} is

$$\mu(x_1)g(x_1)$$

- By integration on x_1, the probability that a particle survives at time $t = 1$ is

$$\int \mu(x_1)g(x_1)dx_1$$

Particle Motion in Random Medium

- ❑ At $t=2$, given the state x_1, the current state x_2 is determined by the transition probability $f\left(x_2 \mid x_1\right)$

- ❑ The probability for such a particle to survive at time $t=2$ is also determined by current state x_2, i.e. the probability is $g(x_2)$

- ❑ If the particle survives at time $t=2$, it means
  1. at time 1, the particle survives with probability $g(x_1)$
  2. state x_1 determines the current state x_2 with probability $f(x_2|x_1)$
  3. the probability to survive at time $t=2$ is $g(x_2)$

The joint probability for the three events is

$$\mu\left(x_1\right) f\left(x_2 \mid x_1\right) g\left(x_1\right) g\left(x_2\right)$$

$\mu\left(x_1\right) f\left(x_2 \mid x_1\right)$ determines the random states

$g\left(x_1\right) g\left(x_2\right)$ determines the probability to survive at each time

Particle Motion in Random Medium

- This can be considered as a typical Hidden Markov Model

Markov Chain (state equation)

$$x_k \sim f\left(x_k \mid x_{k-1}\right)$$

Survive (observation equation)

$$y_k \sim g\left(x_k\right)$$

- The probability density for the particle to survive at time $t = n$ is

$$\mu\left(x_1\right)\prod_{k=2}^{n} f\left(x_k \mid x_{k-1}\right)\cdot\prod_{k=1}^{n} g\left(x_k\right)$$

- By integration over the state variables x_k, we obtain the probability for the particle to survive at time $t = n$

$$\Pr(T > n) = \mathbb{E}_\mu\left[\Pr obability\ of\ \ not\ being\ killed\ given\ X_{1:n}\right] =$$
$$= \int \mu\left(x_1\right)\prod_{k=2}^{n} f\left(x_k \mid x_{k-1}\right)\prod_{k=1}^{n} g\left(x_k\right) d\mathbf{x}_{1:n}$$

Particle Motion in Random Medium

$$\Pr(T > n) = \mathbb{E}_{\mu}\left[\Pr obability\ of\ not\ being\ killed\ given\ X_{1:n}\right] =$$

$$= \int \mu(x_1) \prod_{k=2}^{n} f(x_k \mid x_{k-1}) \prod_{k=1}^{n} g(x_k) d\mathbf{x}_{1:n}$$

- To place this calculation in our SMC framework, we define the following:

$$\gamma_n(\mathbf{x}_{1:n}) = \mu(x_1) \prod_{k=2}^{n} f(x_k \mid x_{k-1}) \cdot \prod_{k=1}^{n} g(x_k)$$

- Then the integration needed to compute the required probability is just the normalization constant of $\gamma_n(x_{1:n})$, i.e.

$$Z_n = \int \mu(x_1) \prod_{k=2}^{n} f(x_k \mid x_{k-1}) \prod_{k=1}^{n} g(x_k) d\mathbf{x}_{1:n}$$

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

$$Z_n = \Pr(T > n)$$

Bayesian Recursion Fomulas for the State Space Model

Bayesian Recursion for the State Space Model

- Let us return to our state space model where the objective is to compute $p(\boldsymbol{x}_{1:n}\,|\,\boldsymbol{y}_{1:n})$. We want to calculate this sequentially.

- We can write the following recursion equation:

$$p(\boldsymbol{x}_{1:n}\,|\,\boldsymbol{y}_{1:n})=\frac{p(\boldsymbol{x}_{1:n},\boldsymbol{y}_{1:n})/p(\boldsymbol{y}_{1:n})}{p(\boldsymbol{x}_{1:n-1},\boldsymbol{y}_{1:n-1})/p(\boldsymbol{y}_{1:n-1})}\,p(\boldsymbol{x}_{1:n-1}\,|\,\boldsymbol{y}_{1:n-1})=\frac{p(\boldsymbol{x}_{1:n},\boldsymbol{y}_{1:n})}{p(\boldsymbol{x}_{1:n-1},\boldsymbol{y}_{1:n-1})}\,\frac{p(\boldsymbol{y}_{1:n-1})}{p(\boldsymbol{y}_{1:n})}\,p(\boldsymbol{x}_{1:n-1}\,|\,\boldsymbol{y}_{1:n-1})$$

$$=g(y_n\,|\,x_n)\,f(x_n\,|\,x_{n-1})\frac{1}{p(y_n\,|\,\boldsymbol{y}_{1:n-1})}\,p(\boldsymbol{x}_{1:n-1}\,|\,\boldsymbol{y}_{1:n-1})=\frac{g(y_n\,|\,x_n)\overbrace{f(x_n\,|\,x_{n-1})\,p(\boldsymbol{x}_{1:n-1}\,|\,\boldsymbol{y}_{1:n-1})}^{\text{Pr}\,edictive:\ p(x_{1:n}|y_{1:n-1})}}{p(y_n\,|\,\boldsymbol{y}_{1:n-1})}$$

where the prediction of y_n given $\boldsymbol{y}_{1:n-1}$ is:

$$p(y_n\,|\,\boldsymbol{y}_{1:n-1})=\int p(y_n,x_n\,|\,\boldsymbol{y}_{1:n-1})\,dx_n=\int g(y_n\,|\,x_n)\,p(x_n\,|\,\boldsymbol{y}_{1:n-1})\,dx_n$$

$$=\int g(y_n\,|\,x_n)\,p(x_n,x_{n-1}\,|\,\boldsymbol{y}_{1:n-1})\,d\boldsymbol{x}_{n-1:n}=\int g(y_n\,|\,x_n)\,f(x_n\,|\,x_{n-1})\,p(x_{n-1}\,|\,\boldsymbol{y}_{1:n-1})\,d\boldsymbol{x}_{n-1:n}$$

- We can write our update equation above in two recursive steps:

$$\textit{Step I - Prediction}:\ p(\boldsymbol{x}_{1:n}\,|\,\boldsymbol{y}_{1:n-1})=f(x_n\,|\,x_{n-1})\,p(\boldsymbol{x}_{1:n-1}\,|\,\boldsymbol{y}_{1:n-1})$$

$$\textit{Step II - Update}:\ p(\boldsymbol{x}_{1:n}\,|\,\boldsymbol{y}_{1:n})=\frac{g(y_n\,|\,x_n)\,p(\boldsymbol{x}_{1:n}\,|\,\boldsymbol{y}_{1:n-1})}{p(y_n\,|\,\boldsymbol{y}_{1:n-1})}\propto g(y_n\,|\,x_n)\,p(\boldsymbol{x}_{1:n}\,|\,\boldsymbol{y}_{1:n-1})$$

Prediction-Updating for the Marginal

- A two-step prediction/update for the marginal (filtering distributions) $p\left(x_n \mid \boldsymbol{y}_{1:n}\right)$ can also be easily derived.

$$\begin{aligned}
\textit{Step I - Prediction}: p\left(x_n \mid \boldsymbol{y}_{1:n-1}\right) &= \int p\left(\boldsymbol{x}_{n-1:n} \mid \boldsymbol{y}_{1:n-1}\right) dx_{n-1} \\
&= \int p\left(x_n \mid x_{n-1}, \boldsymbol{y}_{1:n-1}\right) p\left(x_{n-1} \mid \boldsymbol{y}_{1:n-1}\right) dx_{n-1} \\
&= \int f\left(x_n \mid x_{n-1}\right) p\left(x_{n-1} \mid \boldsymbol{y}_{1:n-1}\right) dx_{n-1}
\end{aligned}$$

$$\textit{Step II - Update}: p\left(x_n \mid \boldsymbol{y}_{1:n}\right) = p\left(x_n \mid y_n, \boldsymbol{y}_{1:n-1}\right) = \frac{g(y_n \mid x_n)\, p\left(x_n \mid \boldsymbol{y}_{1:n-1}\right)}{p\left(y_n \mid \boldsymbol{y}_{1:n-1}\right)}$$

where:

$$p\left(y_n \mid \boldsymbol{y}_{1:n-1}\right) = \int g(y_n \mid x_n) f(x_n \mid x_{n-1}) p\left(x_{n-1} \mid \boldsymbol{y}_{1:n-1}\right) d\boldsymbol{x}_{n-1:n}$$

- This recursion leads to the Kalman filter and the standard HMM filter for linear Gaussian models. In the context of SMC these are not directly useful results.
- Our key emphasis remains in the calculation of $p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)$ even if our interests are in computing $\left\{p\left(x_n \mid \boldsymbol{y}_{1:n}\right)\right\}$

Recursive Calculation of the Marginal $p(\boldsymbol{y}_{1:n})$

- To compute the normalizing factor $p(\boldsymbol{y}_{1:n})$., one can use recursive calculation avoiding high dimensional integration.

$$p(\boldsymbol{y}_{1:n}) = p(y_1)\prod_{k=2}^{n} p(y_k \mid \boldsymbol{y}_{1:k-1})$$

- To compute $p(y_k \mid \boldsymbol{y}_{1:k-1})$, we use the recursion derived earlier:

$$p(y_k \mid \boldsymbol{y}_{1:k-1}) = \int p(y_k, x_k \mid \boldsymbol{y}_{1:k-1}) dx_k = \int g(y_k \mid x_k) p(x_k \mid \boldsymbol{y}_{1:k-1}) dx_k$$

$$= \int g(y_k \mid x_k) p(x_k, x_{k-1} \mid \boldsymbol{y}_{1:k-1}) d\boldsymbol{x}_{k-1:k} = \int g(y_n \mid x_n) f(x_k \mid x_{k-1}) p(x_{k-1} \mid \boldsymbol{y}_{1:k-1}) d\boldsymbol{x}_{k-1:k}$$

- We can now see that the calculation of $p(\boldsymbol{y}_{1:n})$ is a product of lower dimensional integrals.

Forward Filtering Backward Smoothing

- One can also estimate the marginal smoothing distribution $p\left(x_k \mid \boldsymbol{y}_{1:n}\right), k=1,\ldots,n$ (an offline estimate once all measurements $\boldsymbol{y}_{1:n}$ are collected)

$$I\text{ - }Forward\ \ pass: Compute\ and\ store\ p\left(x_k \mid \boldsymbol{y}_{1:k}\right), p\left(x_{k+1} \mid \boldsymbol{y}_{1:k}\right), k=1,2,\ldots,n$$

$$(use\ the\ update\ and\ \ prediction\ recursions\ derived\ earlier)$$

$$II\text{ - }Backward\ \ pass\ (k=n\text{-}1, n\text{-}2,..,1): p\left(x_k \mid \boldsymbol{y}_{1:n}\right)=\int \frac{f\left(x_{k+1} \mid x_k\right) p\left(x_k \mid \boldsymbol{y}_{1:k}\right)}{p\left(x_{k+1} \mid \boldsymbol{y}_{1:k}\right)} p\left(x_{k+1} \mid \boldsymbol{y}_{1:n}\right) d x_{k+1}$$

- Indeed, one can show:

$$p\left(x_k \mid \boldsymbol{y}_{1:n}\right)=\int p\left(x_k, x_{k+1} \mid \boldsymbol{y}_{1:n}\right) d x_{k+1}=\int p\left(x_k \mid x_{k+1}, \boldsymbol{y}_{1:n}\right) p\left(x_{k+1} \mid \boldsymbol{y}_{1:n}\right) d x_{k+1}$$

$$=\int p\left(x_k \mid x_{k+1}, \boldsymbol{y}_{1:k}\right) p\left(x_{k+1} \mid \boldsymbol{y}_{1:n}\right) d x_{k+1}=\int \frac{f\left(x_{k+1} \mid x_k\right) p\left(x_k \mid \boldsymbol{y}_{1:k}\right)}{p\left(x_{k+1} \mid \boldsymbol{y}_{1:k}\right)} p\left(x_{k+1} \mid \boldsymbol{y}_{1:n}\right) d x_{k+1}$$

- Here we used (see Appendix next) $p\left(x_k \mid x_{k+1}, \boldsymbol{y}_{1:n}\right)=p\left(x_k \mid x_{k+1}, \boldsymbol{y}_{1:k}\right)$

Appendix

- ❑ Here we prove the Eq. used in the earlier slide:

$$p(x_k \mid x_{k+1}, \boldsymbol{y}_{1:n}) = p\left(x_k \mid x_{k+1}, \boldsymbol{y}_{1:k}\right)$$

- ❑ Note that:

$$p\left(x_k \mid x_{k+1}, \boldsymbol{y}_{1:\mathrm{n}}\right) = \frac{p\left(x_k, x_{k+1}, \boldsymbol{y}_{1:\mathrm{n}}\right)}{p\left(x_{k+1}, \boldsymbol{y}_{1:\mathrm{n}}\right)} = \frac{\int p\left(\boldsymbol{x}_{1:\mathrm{n}}, \boldsymbol{y}_{1:\mathrm{n}}\right) d\boldsymbol{x}_{1:k-1} d\boldsymbol{x}_{k+2:n}}{\int p\left(\boldsymbol{x}_{1:\mathrm{n}}, \boldsymbol{y}_{1:\mathrm{n}}\right) d\boldsymbol{x}_{1:k} d\boldsymbol{x}_{k+2:n}}$$

$$= \frac{\int p\left(x_1\right) \prod_{i=1}^{n-1} f\left(x_{i+1} \mid x_i\right) g\left(y_i \mid x_i\right) g\left(y_n \mid x_n\right) d\boldsymbol{x}_{1:k-1} d\boldsymbol{x}_{k+2:n}}{\int p\left(x_1\right) \prod_{i=1}^{n-1} f\left(x_{i+1} \mid x_i\right) g\left(y_i \mid x_i\right) g\left(y_n \mid x_n\right) d\boldsymbol{x}_{1:k} d\boldsymbol{x}_{k+2:n}}$$

$$= \frac{\int p\left(x_1\right) \prod_{i=1}^{k} f\left(x_{i+1} \mid x_i\right) g\left(y_i \mid x_i\right) d\boldsymbol{x}_{1:k-1} \cancel{\int \prod_{i=k+1}^{n-1} f\left(x_{i+1} \mid x_i\right) g\left(y_i \mid x_i\right) g\left(y_n \mid x_n\right) d\boldsymbol{x}_{k+2:n}}}{\int p\left(x_1\right) \prod_{i=1}^{k} f\left(x_{i+1} \mid x_i\right) g\left(y_i \mid x_i\right) d\boldsymbol{x}_{1:k} \cancel{\int \prod_{i=k+1}^{n-1} f\left(x_{i+1} \mid x_i\right) g\left(y_i \mid x_i\right) g\left(y_n \mid x_n\right) d\boldsymbol{x}_{k+2:n}}}$$

$$= \frac{p\left(x_k, x_{k+1}, \boldsymbol{y}_{1:k}\right)}{p\left(x_{k+1}, \boldsymbol{y}_{1:k}\right)} = p\left(x_k \mid x_{k+1}, \boldsymbol{y}_{1:\mathrm{k}}\right)$$

Forward-Backward (Two-Filter) Smoother

- One can also estimate the marginal smoothing distribution as follows (see proof on the following slide):

$$p\left(x_k \mid \boldsymbol{y}_{1:n}\right), k=1,\dots,n$$

$$\begin{aligned}
&\textit{Step I - Backward information filter}: p\left(\boldsymbol{y}_{k+1:n} \mid x_k\right) = \int p\left(\boldsymbol{y}_{k+1:n}, x_{k+1} \mid x_k\right) dx_{k+1} \\
&= \int p\left(\boldsymbol{y}_{k+1:n} \mid x_{k+1}, x_k\right) f\left(x_{k+1} \mid x_k\right) dx_{k+1} \\
&= \int p\left(\boldsymbol{y}_{k+2:n} \mid x_{k+1}\right) g(y_{k+1} \mid x_{k+1}) f\left(x_{k+1} \mid x_k\right) dx_{k+1} \\
&\textit{Step II - Update}: p\left(x_k \mid \boldsymbol{y}_{1:n}\right) = \frac{p(x_k \mid \boldsymbol{y}_{1:k}) p\left(\boldsymbol{y}_{k+1:n} \mid x_k\right)}{p\left(\boldsymbol{y}_{k+1:n} \mid \boldsymbol{y}_{1:k}\right)}
\end{aligned}$$

- Note that we can have: $\int p\left(\boldsymbol{y}_{k+1:n} \mid x_k\right) dx_k = \infty$. This can lead to wrong algorithms.

- This is known as the forward-backward smoother.

Proof of the Two-Filter Smoother

- Note that: $p\left(\boldsymbol{y}_{k+1:n} \mid x_k, \boldsymbol{y}_{1:k}\right) = p\left(\boldsymbol{y}_{k+1:n} \mid x_k\right)$. We can look at each term separately:

$$p\left(\boldsymbol{y}_{k+1:n} \mid x_k, \boldsymbol{y}_{1:k}\right) = \frac{p\left(\boldsymbol{y}_{1:n}, x_k\right)}{p\left(x_k, \boldsymbol{y}_{1:k}\right)} = \frac{\int p\left(x_1\right) g\left(y_n \mid x_n\right) \prod_{i=1}^{n-1} f\left(x_{i+1} \mid x_i\right) g\left(y_i \mid x_i\right) d\boldsymbol{x}_{1:k-1} d\boldsymbol{x}_{k+1:n}}{\int p\left(x_1\right) g\left(y_k \mid x_k\right) \prod_{i=1}^{k-1} f\left(x_{i+1} \mid x_i\right) g\left(y_i \mid x_i\right) d\boldsymbol{x}_{1:k-1}}$$

$$= \frac{\int p\left(x_1\right) g\left(y_k \mid x_k\right) \prod_{i=1}^{k-1} f\left(x_{i+1} \mid x_i\right) g\left(y_i \mid x_i\right) d\boldsymbol{x}_{1:k-1} \int f\left(x_{k+1} \mid x_k\right) g\left(y_n \mid x_n\right) \prod_{i=k+1}^{n-1} f\left(x_{i+1} \mid x_i\right) g\left(y_i \mid x_i\right) d\boldsymbol{x}_{k+1:n}}{\int p\left(x_1\right) g\left(y_k \mid x_k\right) \prod_{i=1}^{k-1} f\left(x_{i+1} \mid x_i\right) g\left(y_i \mid x_i\right) d\boldsymbol{x}_{1:k-1}}$$

$$= \int f\left(x_{k+1} \mid x_k\right) g\left(y_n \mid x_n\right) \prod_{i=k+1}^{n-1} f\left(x_{i+1} \mid x_i\right) g\left(y_i \mid x_i\right) d\boldsymbol{x}_{k+1:n}$$

$$p\left(\boldsymbol{y}_{k+1:n} \mid x_k\right) = \frac{p\left(\boldsymbol{y}_{k+1:n}, x_k\right)}{p\left(x_k\right)} = \frac{\int p\left(x_k\right) f\left(x_{k+1} \mid x_k\right) g\left(y_n \mid x_n\right) \prod_{i=k+1}^{n-1} f\left(x_{i+1} \mid x_i\right) g\left(y_i \mid x_i\right) d\boldsymbol{x}_{k+1:n}}{p\left(x_k\right)}$$

$$= \int f\left(x_{k+1} \mid x_k\right) g\left(y_n \mid x_n\right) \prod_{i=k+1}^{n-1} f\left(x_{i+1} \mid x_i\right) g\left(y_i \mid x_i\right) d\boldsymbol{x}_{k+1:n}$$

- The update rule is then:

$$p\left(x_k \mid \boldsymbol{y}_{1:n}\right) = p\left(x_k \mid \boldsymbol{y}_{1:k}, \boldsymbol{y}_{k+1:n}\right) = \frac{p\left(x_k, \boldsymbol{y}_{k+1:n} \mid \boldsymbol{y}_{1:k}\right)}{p\left(\boldsymbol{y}_{k+1:n} \mid \boldsymbol{y}_{1:k}\right)}$$

$$= \frac{p\left(\boldsymbol{y}_{k+1:n} \mid x_k, \boldsymbol{y}_{1:k}\right) p\left(x_k \mid \boldsymbol{y}_{1:k}\right)}{p\left(\boldsymbol{y}_{k+1:n} \mid \boldsymbol{y}_{1:k}\right)} = \frac{p\left(\boldsymbol{y}_{k+1:n} \mid x_k\right) p\left(x_k \mid \boldsymbol{y}_{1:k}\right)}{p\left(\boldsymbol{y}_{k+1:n} \mid \boldsymbol{y}_{1:k}\right)}$$

Bayesian Recursion for the State Space Model

- Let us return back to our main objective: computing $p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)$

$$\text{Step I - Prediction}: p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n-1}\right) = f\left(x_n \mid x_{n-1}\right) p\left(\boldsymbol{x}_{1:n-1} \mid \boldsymbol{y}_{1:n-1}\right)$$

$$\text{Step II - Update}: p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right) = \frac{g(y_n \mid x_n) p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n-1}\right)}{p\left(y_n \mid \boldsymbol{y}_{1:n-1}\right)} \propto g(y_n \mid x_n) p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n-1}\right)$$

- We will apply sequential Monte Carlo methods to approximate the target distribution.

Online Bayesian 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 $Z_n = (Xn, \theta_n)$.

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

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

Maximum Likelihood Parameter Estimation

- ❑ Standard approaches for parameter estimation consists of computing the Maximum Likelihood (ML) estimate

$$\theta_{ML} = \arg\max \log p_\theta\left(\boldsymbol{y}_{1:n}\right)$$

- ❑ The likelihood function can be multimodal and there is no guarantee to find its global optimum.

- ❑ Standard (stochastic) gradient algorithms can be used (e.g. based on Fisher's identity) to find a local minimum:

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

- ❑ These algorithms can work decently but it can be difficult to scale the components of the gradients.

- ❑ Note that these algorithms involve computing $p_\theta\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)$ which is one of our main SMC algorithmic results.

Expectation/Maximization for HMM

- One can also use the EM algorithm

$$\theta^{(i)} = Q\left(\theta^{(i)},\theta\right)$$

$$\begin{aligned} Q\left(\theta^{(i)},\theta\right) &= \int \log p_{\theta}\left(\boldsymbol{x}_{1:n},\boldsymbol{y}_{1:n}\right) p_{\theta^{(i-1)}}\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right) d\boldsymbol{x}_{1:n} \\ &= \int \log\left(\mu\left(x_1\right) g\left(y_1 \mid x_1\right)\right) p_{\theta^{(i-1)}}\left(x_1 \mid \boldsymbol{y}_{1:n}\right) dx_1 \\ &+ \sum_{k=2}^{n} \int \log\left(f\left(x_k \mid x_{k-1}\right) g\left(y_k \mid x_k\right)\right) p_{\theta^{(i-1)}}\left(\boldsymbol{x}_{k-1:k} \mid \boldsymbol{y}_{1:n}\right) d\boldsymbol{x}_{k-1:k} \end{aligned}$$

- Above we used:

$$p\left(\boldsymbol{x}_{1:n},\boldsymbol{y}_{1:n}\right) = \mu\left(x_1\right) \prod_{k=2}^{n} f\left(x_k \mid x_{k-1}\right) \prod_{k=1}^{n} g\left(y_k \mid x_k\right)$$

- Implementation of the EM algorithm requires computing expectations with respect to the smoothing distributions $p_{\theta^{(i-1)}}\left(\boldsymbol{x}_{k-1:k} \mid \boldsymbol{y}_{1:n}\right)$

Closed Form Inference in HMM

- We have closed-form solutions for finite state-space HMM as all integrals are becoming finite sums

- Linear Gaussian models; all the posterior distributions are Gaussian; (Kalman filter).

- In most cases of interest, it is not possible to compute the solution in closed-form and we need numerical approximations.

- This is the case for all non-linear non-Gaussian models.

- SMC methods for such problems are in some sense asymptotically consistent.

Closed Form Inference in HMM

- ❑ Gaussian approximations: Extended Kalman filter, Unscented Kalman filter.

- ❑ Gaussian sum approximations.

- ❑ Projection filters (similar to Variational methods in machine learning).

- ❑ Simple discretization of the state-space.

- ❑ Analytical methods work in simple cases but are not reliable and it is difficult to diagnose when they fail.

- ❑ Standard discretization of the space is expensive and difficult to implement in high-dimensional scenarios.

- ❑ We need numerical approximations.

Importance Sampling and its Application to Nonlinear Non-Gaussian Dynamic Models

Review: Importance Sampling

- Our goal is to compute an expectation value of the form :

$$\mathbb{E}_{\pi}\left[f(\boldsymbol{x})\right] = \int_A f(\boldsymbol{x})\pi(\boldsymbol{x})d\boldsymbol{x}$$

where $\pi(\boldsymbol{x})$ is a probability distribution (posterior inference in Bayesian models, Bayesian model validation, etc.)

- We assume that $\pi(x) = \frac{\gamma(x)}{Z}$ where $Z = \int \gamma(x)dx$ is unknown and γ is known pointwise.

- The basic idea in Monte Carlo methods is to sample N i.i.d. random numbers $\boldsymbol{X}^{(i)} \overset{i.i.d.}{\sim} \pi(.)$ and build an empirical measure

$$\hat{\pi}(\boldsymbol{x})d\boldsymbol{x} = \frac{1}{N}\sum_{i=1}^{N} \delta_{\boldsymbol{X}^{(i)}}\, d\boldsymbol{x}$$

- Using this:

$$\mathbb{E}_{\hat{\pi}}[f(\boldsymbol{x})] = \frac{1}{N}\sum_{i=1}^{N} f\left(\boldsymbol{X}^{(i)}\right) \;,\; where\, \boldsymbol{X}^{(i)} \overset{i.i.d.}{\sim} \pi(.)$$

J.S. Liu, Monte Carlo Strategies in Scientific Computing, Chapter 3, Springer-Verlag, New York.

Monte Carlo Methods

- Using the approximation of π:

$$\mathbb{E}_{\hat{\pi}}[f(\boldsymbol{x})] = \frac{1}{N}\sum_{i=1}^{N} f\left(\boldsymbol{X}^{(i)}\right) \quad , \quad where\, \boldsymbol{X}^{(i)} \overset{i.i.d.}{\sim} \pi(.)$$

- The following hold:

$$\mathbb{E}[\mathbb{E}_{\hat{\pi}}(f)] = \mathbb{E}_{\pi}(f), V[\mathbb{E}_{\hat{\pi}}(f)] = \frac{1}{N}\mathbb{E}_{\pi}\left(\left(f-\mathbb{E}_{\pi}(f)\right)^2\right), \sqrt{N}\left(\mathbb{E}_{\hat{\pi}}(f) - \mathbb{E}_{\pi}(f)\right) \sim \mathcal{N}\left(0, \mathbb{E}_{\pi}\left(\left(f-\mathbb{E}_{\pi}(f)\right)^2\right)\right)$$

- Similarly, marginalization is also simple:

$$\hat{\pi}(x_p)dx_p = \int \hat{\pi}(x_1, x_2, \ldots, x_n)\, d\boldsymbol{x}_{1:p-1}\, d\boldsymbol{x}_{p+1:n} = \frac{1}{N}\sum_{i=1}^{N} \delta_{X_p^{(i)}}\, dx_p$$

- In MC, the samples automatically concentrate in regions of high probability mass regardless of the dimension of the space.

- However, it is not always easy or effective to sample from the original probability distribution $\pi(\boldsymbol{x})$. A more effective strategy is to focus on the regions of "importance" in $\pi(\boldsymbol{x})$ so as to save computational resources.

J.S. Liu, Monte Carlo Strategies in Scientific Computing, Chapter 3, Springer-Verlag, New York.

Review: Importance Sampling

- We assume that $\pi(\boldsymbol{x})$ is only known up to a normalizing constant:

$$\pi(\boldsymbol{x}) = \frac{\gamma(\boldsymbol{x})}{Z}$$

- For any distribution $q(\boldsymbol{x})$ such that $\pi(\boldsymbol{x}) > 0 \Rightarrow q(\boldsymbol{x}) > 0$, we can write:

$$\pi(\boldsymbol{x}) = \frac{w(\boldsymbol{x})q(\boldsymbol{x})}{\underbrace{\int w(\boldsymbol{x})q(\boldsymbol{x})dx}_{Z}} = \frac{w(\boldsymbol{x})q(\boldsymbol{x})}{Z}, \text{ where } w(\boldsymbol{x}) = \frac{\gamma(\boldsymbol{x})}{q(\boldsymbol{x})}$$

- The proposal distribution $q(\boldsymbol{x})$ is known as "importance density" or "trial density". $w(\boldsymbol{x})$ is called the importance weight.

- The importance density can be chosen arbitrarily as any proposal easy to sample from:

$$\boldsymbol{X}^{(i)} \overset{i.i.d.}{\sim} q(\boldsymbol{x}) \Rightarrow \widehat{\boldsymbol{q}}(\boldsymbol{x})d\boldsymbol{x} = \frac{1}{N}\sum_{i=1}^{N} \delta_{\boldsymbol{X}^{(i)}}(d\boldsymbol{x})$$

Review: Importance Sampling

- Substitution of $\hat{q}(\boldsymbol{x})d\boldsymbol{x} = \frac{1}{N}\sum_{i=1}^{N}\delta_{\boldsymbol{X}^{(i)}}(d\boldsymbol{x})$ in the importance sampling identity gives:

$$\hat{\pi}(\boldsymbol{x})d\boldsymbol{x} = \frac{w(\boldsymbol{x})\hat{q}(\boldsymbol{x})}{\int w(\boldsymbol{x})\hat{q}(\boldsymbol{x})d\boldsymbol{x}}d\boldsymbol{x} = \frac{\frac{1}{N}\sum_{i=1}^{N} w\left(\boldsymbol{X}^{(i)}\right)\delta_{\boldsymbol{X}^{(i)}}(d\boldsymbol{x})}{\frac{1}{N}\sum_{i=1}^{N} w\left(\boldsymbol{X}^{(i)}\right)} = \sum_{i=1}^{N} W^{(i)}\,\delta_{\boldsymbol{X}^{(i)}}(d\boldsymbol{x}),$$

$$where\ W^{(i)} \propto w\left(\boldsymbol{X}^{(i)}\right)\ and\ \sum_{i=1}^{N} W^{(i)} = 1$$

- Similarly, we can approximate the normalization factor of our target distribution as follows:

$$\hat{Z} = \int \frac{\gamma(\boldsymbol{x})}{q(\boldsymbol{x})}\hat{q}(\boldsymbol{x})d\boldsymbol{x} = \int w(\boldsymbol{x})\hat{q}(\boldsymbol{x})d\boldsymbol{x} = \frac{1}{N}\sum_{i=1}^{N} w\left(\boldsymbol{X}^{(i)}\right) = \frac{1}{N}\sum_{i=1}^{N}\frac{\gamma\left(\boldsymbol{X}^{(i)}\right)}{q\left(\boldsymbol{X}^{(i)}\right)}$$

Review: Importance Sampling

$$\hat{\pi}(\boldsymbol{x})d\boldsymbol{x} = \sum_{i=1}^{N} W^{(i)}\,\delta_{\boldsymbol{X}^{(i)}}(d\boldsymbol{x}),\ where\, W^{(i)} \propto w\left(\boldsymbol{X}^{(i)}\right)\ and\ \sum_{i=1}^{N} W^{(i)} = 1$$

- ❑ The distribution $\pi(\boldsymbol{x})$ is now approximated by a weighted sum of delta masses, where the weights compensate for the discrepancy between $\pi(\boldsymbol{x})$ and $q(\boldsymbol{x})$.

Review: Importance Sampling

- Similarly calculation of $\mathbb{E}_{\pi}\left[f(x)\right]$ using importance sampling gives:

$$\mathbb{E}_{\hat{\pi}}[f(\boldsymbol{x})] = \int_A f(\boldsymbol{x})\hat{\pi}(\boldsymbol{x})d\boldsymbol{x} = \sum_{i=1}^{N} f\left(\boldsymbol{X}^{(i)}\right) W^{(i)}$$

- The statistics of this estimate are given for $N >> 1$ as follows:

$$\mathbb{E}\left[\mathbb{E}_{\hat{\pi}}[f(\boldsymbol{x})]\right] = \mathbb{E}_{\pi}[f(\boldsymbol{x})] - \frac{1}{N_{\pi}}\mathbb{E}[W(\boldsymbol{X})(f(\boldsymbol{X}) - \mathbb{E}_{\pi}[f(\boldsymbol{x})])]$$

$$V\left[\mathbb{E}_{\hat{\pi}}[f(\boldsymbol{x})]\right] = \frac{1}{N_{\pi}}\mathbb{E}[W(\boldsymbol{X})(f(\boldsymbol{X}) - \mathbb{E}_{\pi}[f(\boldsymbol{x})])^2]$$

where as you recall we have some negligible bias:

$$\frac{1}{N_{\pi}}\mathbb{E}[W(\boldsymbol{X})(f(\boldsymbol{X}) - \mathbb{E}_{\pi}[f(\boldsymbol{x})])]$$

Statistics of the Normalization Constant

- We can similarly compute the statistics of the normalization constant:

$$\hat{Z} = \int \frac{\gamma(\boldsymbol{x})}{q(\boldsymbol{x})}\hat{q}(\boldsymbol{x})d\boldsymbol{x} = \frac{1}{N}\sum_{i=1}^{N}\frac{\gamma\left(\boldsymbol{X}^{(i)}\right)}{q\left(\boldsymbol{X}^{(i)}\right)} = \frac{1}{N}\sum_{i=1}^{N} w\left(\boldsymbol{X}^{(i)}\right)$$

- They are given as:

$$\mathbb{E}\left[\hat{Z}\right] = Z, \; and$$

$$V\left[\hat{Z}\right] = \frac{1}{N}\mathbb{E}_q\left[\left(\frac{\gamma(\boldsymbol{x})}{q(\boldsymbol{x})} - Z\right)^2\right]$$

Review: Importance Sampling

- We select $q(\boldsymbol{x})$ *as close as possible to* $\pi(\boldsymbol{x})$.

- The variance of the weights is bounded iff

$$\int \frac{\gamma^2(\boldsymbol{x})}{q(\boldsymbol{x})} d\boldsymbol{x} < \infty$$

- In practice, it is sufficient to ensure that the weights are bounded:

$$w(\boldsymbol{x}) = \frac{\gamma(\boldsymbol{x})}{q(\boldsymbol{x})} < \infty$$

- This is equivalent to saying that $q(\boldsymbol{x})$ should have heavier tails than $\pi(\boldsymbol{x})$.

Monte Carlo for the State Space Model

- We are interested to estimate the high-dimensional density

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

- For now let us start with a fixed n.

- A Monte Carlo approximation (empirical measure) of our target distribution is of the form:

$$\hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n}), \; where \, \boldsymbol{X}_{1:n}^{(i)} \sim p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})$$

- For any function $\varphi\left(\boldsymbol{x}_{1:n}\right) : \mathcal{X}^n \to \mathbb{R}$, we can use a Monte Carlo approximation of its expectation as:

$$\mathbb{E}_{\hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi) = \int_{\mathcal{X}^n} \varphi(\boldsymbol{x}_{1:n})\, \hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) d\boldsymbol{x}_{1:n}$$

$$= \int_{\mathcal{X}^n} \varphi(\boldsymbol{x}_{1:n}) \frac{1}{N}\sum_{i=1}^{N} \delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n}) d\boldsymbol{x}_{1:n} = \frac{1}{N}\sum_{i=1}^{N} \varphi\left(\boldsymbol{X}_{1:n}^{(i)}\right)$$

Monte Carlo for the State Space Model

- This earlier estimate is asymptotically consistent (converges towards $\mathbb{E}_{p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi)$).

- The estimate is unbiased and its variance gives the following convergence properties:

$$Var_{\boldsymbol{X}_{1:n}^{(i)}}\left[\mathbb{E}_{\hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi)\right] = \frac{1}{N} Var_{p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi)$$

$$\sqrt{N}\left(\mathbb{E}_{\hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi) - \mathbb{E}_{p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi)\right) \overset{d}{\rightarrow} \mathcal{N}\left(0, Var_{p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi)\right)$$

- The rate of convergence is independent of n. This does not imply that Monte Carlo bits the curse of dimensionality since it is possible that $Var_{p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi)$ increases (with time) n.

Monte Carlo for the State Space Model

- ❑ The Monte Carlo approximation can easily be used to compute any marginal distribution, e.g. $p\left(x_k \mid \boldsymbol{y}_{1:n}\right)$

$$\hat{p}_N(x_k|\boldsymbol{y}_{1:n}) = \int_{\chi^{n-1}} \hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})d\boldsymbol{x}_{1:k-1}d\boldsymbol{x}_{k+1:n}$$

$$= \int_{\chi^{n-1}} \frac{1}{N}\sum_{i=1}^{N} \delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n})d\boldsymbol{x}_{1:k-1}d\boldsymbol{x}_{k+1:n}$$

$$= \frac{1}{N}\sum_{i=1}^{N} \delta_{X_k^{(i)}}(x_k)$$

- ❑ Note that the marginal likelihood $p\left(\boldsymbol{y}_{1:n}\right)$ cannot be estimated as easily using $\boldsymbol{X}_{1:n}^{(i)} \sim p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)$

Difficulties with Standard Monte Carlo Sampling

- It is difficult to sample from our target high-dimensional distribution:

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

- MCMC methods are not useful in this context.

- As n increases, we would like to be able to sample from $p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)$ with an algorithm that keeps the computational cost fixed at each time step n.

Importance Sampling for our State Space Model

- Rather than sampling directly from our target distribution $p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$, we should sample from an importance distribution $q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$

- Note that in the notation here for q, $\boldsymbol{y}_{1:n}$ is used as a parameter – not to indicate any posterior distribution.

- The importance distribution needs to satisfy the following properties:

  - The support of $q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$ includes the support of $p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$ i.e.

$$p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}) > 0 \Rightarrow q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}) > 0$$

  - It is easy to sample from $q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$

- We use the following identity:

$$p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}) = \frac{p(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n})}{\int p(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n})\, d\boldsymbol{x}_{1:n}} = \frac{\left[p(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}) / q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})\right] q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})}{\int \left[p(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}) / q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})\right] q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})\, d\boldsymbol{x}_{1:n}}$$

$$= \frac{w(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n})\, q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})}{\int w(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n})\, q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})\, d\boldsymbol{x}_{1:n}}$$

Importance Sampling for our State Space Model

- ❑ Let us draw N samples from our importance distribution:

$$\boldsymbol{X}_{1:n}^{(i)} \sim q(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}),\ \hat{q}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) = \frac{1}{N}\sum_{i=1}^{N}\delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n})$$

- ❑ Then using the identity in the earlier slide, we obtain the following approximation of our target distribution:

$$\hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) = \frac{w(\boldsymbol{x}_{1:n},\boldsymbol{y}_{1:n})\hat{q}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}{\int w(\boldsymbol{x}_{1:n},\boldsymbol{y}_{1:n})\hat{q}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})d\boldsymbol{x}_{1:n}}$$

$$= \frac{w(\boldsymbol{x}_{1:n},\boldsymbol{y}_{1:n})\frac{1}{N}\sum_{i=1}^{N}\delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n})}{\int w(\boldsymbol{x}_{1:n},\boldsymbol{y}_{1:n})\frac{1}{N}\sum_{i=1}^{N}\delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n})d\boldsymbol{x}_{1:n}}$$

$$= \sum_{i=1}^{N} W_n^{(i)}\delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n}),\ W_n^{(i)} = \frac{w\left(\boldsymbol{X}_{1:n}^{(i)},\boldsymbol{y}_{1:n}\right)}{\sum_{i=1}^{N} w\left(\boldsymbol{X}_{1:n}^{(i)},\boldsymbol{y}_{1:n}\right)}$$

- ❑ Note that: $\hat{p}_N(\boldsymbol{y}_{1:n}) = \int w(\boldsymbol{x}_{1:n},\boldsymbol{y}_{1:n})\frac{1}{N}\sum_{i=1}^{N}\delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n})d\boldsymbol{x}_{1:n} = \frac{1}{N}\sum_{i=1}^{N} w\left(\boldsymbol{X}_{1:n}^{(i)},\boldsymbol{y}_{1:n}\right)$

Normalized Weights in Importance Sampling

- We defined earlier the unnormalized weights as follows:

$$Unnormalized\ weights: w\left(\boldsymbol{x}_{1:n},\boldsymbol{y}_{1:n}\right)=\frac{p\left(\boldsymbol{x}_{1:n},\boldsymbol{y}_{1:n}\right)}{q\left(\boldsymbol{x}_{1:n}\mid\boldsymbol{y}_{1:n}\right)}=p\left(\boldsymbol{y}_{1:n}\right)\underbrace{\frac{p\left(\boldsymbol{x}_{1:n}\mid\boldsymbol{y}_{1:n}\right)}{q\left(\boldsymbol{x}_{1:n}\mid\boldsymbol{y}_{1:n}\right)}}_{\substack{\text{Discrepancy between}\\ \text{target distribution and}\\ \text{importance distribution}}}$$

- The normalized weights were also introduced as:

$$Normalized\ weights: W_n^{(i)}=\frac{w\left(\boldsymbol{X}_{1:n}^{(i)},\boldsymbol{y}_{1:n}\right)}{\sum_{i=1}^{N}w\left(\boldsymbol{X}_{1:n}^{(i)},\boldsymbol{y}_{1:n}\right)}$$

Optimal Importance Sampling Distribution

- ❑ $\hat{p}_N(\boldsymbol{y}_{1:n})$ is an unbiased estimate of $p(\boldsymbol{y}_{1:n})$ with variance:

$$\frac{1}{N}\left[\int w^2(\boldsymbol{x}_{1:n},\boldsymbol{y}_{1:n})\,q(\boldsymbol{x}_{1:n}\mid\boldsymbol{y}_{1:n})\,d\boldsymbol{x}_{1:n}-1\right]$$

- ❑ You can bring this variance to zero with the selection

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

  Of course this is what we wanted to avoid (we want to sample from an easier distribution).

- ❑ However, this results points to the fact that the choice of q needs to be as close as possible to the target distribution.

Importance Sampling Estimates

- We are interested in an importance sampling approximation of $\mathbb{E}_{p\left(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}\right)}\left(\varphi\right)$.

$$\mathbb{E}_{\hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi) = \sum_{i=1}^{N} W_n^{(i)} \varphi\left(\boldsymbol{X}_{1:n}^{(i)}\right)$$

- This is a biased estimate for a finite N and we have shown in our earlier lecture on Importance Sampling that:

$$\lim_{N\to\infty} N\left[\mathbb{E}_{p_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi) - \mathbb{E}_{p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi)\right] = -\int \frac{p^2(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}{q(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}\left(\varphi(\boldsymbol{x}_{1:n}) - \mathbb{E}_{p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi)\right) d\boldsymbol{x}_{1:n}$$

$$\sqrt{N}\left(\mathbb{E}_{p_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi) - \mathbb{E}_{p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi)\right) \xrightarrow{d} \mathcal{N}\left(0, \int \frac{p^2(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}{q(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}\left(\varphi(\boldsymbol{x}_{1:n}) - \mathbb{E}_{p(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n})}(\varphi)\right)^2 d\boldsymbol{x}_{1:n}\right)$$

- The asymptotic bias is of the order $1/N$ (negligible) and the MSE error is:

$$MSE = \underset{O\left(N^{-2}\right)}{bias^2} + \underbrace{variance}_{O\left(N^{-1}\right)}$$

Selection of Importance Sampling Distribution

- ❑ As discussed before, the importance sampling distribution should be selected so that the weights are bounded or equivalently $q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$ has heavier tails than $p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$

$$w(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}) \leq C \ \forall \boldsymbol{x}_{1:n} \in \mathcal{X}^n$$

- ❑ To minimize the asymptotic bias, we aim for $q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$ that is as close as possible to $p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$

- ❑ Note that the selection of the importance sampling needs to be not only such that it covers the support of the target but also needs to be a clever one for the particular problem of interest.

- ❑ For numerical examples and MatLab implementations please see an earlier lecture on importance sampling.

Effective Sample Size

- In our importance sampling approximation from the target $p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$ using the importance distribution $q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$ (for a fixed n), we would like ideally to have

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

- In this case, all the unnormalized importance weights will be equal and their variance equal to zero.

- To access the quality of the importance sampling approximation, note that for flat functions,

$$\frac{\textit{Variance of IS estimate}}{\textit{Variance of Standard MC estimate}} \approx 1 + Var_{q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})} W(\boldsymbol{X}_{1:n} \mid \boldsymbol{y}_{1:n})$$

- This is often interpreted as the effective sample size (N weighted samples from $q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$ are approximately equivalent to M unweighted samples from $p(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})$)

$$M = \frac{N}{1 + Var_{q(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n})} W(\boldsymbol{X}_{1:n} \mid \boldsymbol{y}_{1:n})} \leq N$$

Effective Sample Size

- ❑ We often approximate the effective sample size M as follows:

$$ESS = \left(\sum_{i=1}^{N} W_n^{(i)2} \right)^{-1}$$

since

$$Var_{q\left(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}\right)} W\left(\boldsymbol{X}_{1:n}^{(i)} \mid \boldsymbol{y}_{1:n}\right) \approx N \sum_{i=1}^{N} W^2\left(\boldsymbol{X}_{1:n}^{(i)} \mid \boldsymbol{y}_{1:n}\right) - 1$$

- ❑ We clearly can see that

$$1 \leq ESS = \left(\sum_{i=1}^{N} w_n^{(i)2} \right)^{-1} \leq N$$

- ❑ We can thus have

  - $ESS = 1$ (one of the weights equal to 1, all other zero, very inefficient) to

  - $ESS = N$ (all weights equal to $1/N$, excellent sampling).

Sequential Importance Sampling

- Let us return to our state space model and consider a sequential Monte Carlo approximation of $p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right) \propto p\left(\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}\right)$

- The distributions $\left\{\pi_n = p\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)\right\}$ are known up to a normalizing constant:

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

- We want to estimate the expectations of functions $f_n : \mathscr{X}^n \to \mathbb{R}$

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

and/or the normalizing constants Z_n.

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

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

Sequential Importance Sampling

- 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.

Sequential Importance Sampling

- We want to design a sequential 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 $\hat{\pi}_1(x_1) = \hat{p}_N(x_1, y_1), \hat{Z}_1$ using an importance density $q_1(x_1 \mid y_1)$.

$$X_1^{(i)} \sim q_1(x_1|y_1),\ i = 1,2,\ldots,N$$

$$\hat{p}_N(x_1, y_1)dx_1 = \sum_{i=1}^{N} W_1^{(i)}\, \delta_{X_1^{(i)}}(dx_1),\ where\ W_1^{(i)} = \frac{w_1\left(X_1^{(i)}, y_1\right)}{\sum_{j=1}^{N} w_1\left(X_1^{(j)}, y_1\right)}$$

$$\hat{Z}_1 = \frac{1}{N}\sum_{i=1}^{N} w_1\left(X_1^{(i)}, y_1\right)\ with$$

$$w_1(x_1, y_1) = \frac{\gamma_1(x_1)}{q_1(x_1|y_1)} = \frac{p(x_1, y_1)}{q_1(x_1|y_1)}$$

Sequential Importance Sampling

- At `time 2', we want to approximate $\hat{\pi}_2(\boldsymbol{x}_{1:2}) = \hat{p}_N(\boldsymbol{x}_{1:2}, \boldsymbol{y}_{1:2})$, $\hat{Z}_2$ using an importance density $q_2(\boldsymbol{x}_{1:2} \mid \boldsymbol{y}_{1:2})$.

- We want to reuse the samples $\boldsymbol{X}_1^{(i)}$ and $q_1(x_1|y_1)$ in building the importance sampling approximation for $\pi_2(\boldsymbol{x}_{1:2}), Z_2$.

- Let us select $q_2(\boldsymbol{x}_{1:2} \mid \boldsymbol{y}_{1:2}) = q_1(x_1 \mid y_1)\, q_2(x_2 \mid \boldsymbol{y}_{1:2}, x_1)$

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

$$X_2^{(i)} \mid X_1^{(i)} \sim q_2\left(x_2 \mid \boldsymbol{y}_{1:2}, X_1^{(i)}\right)$$

- The importance sampling weight for this step is then:

$$w_2(\boldsymbol{x}_{1:2}, \boldsymbol{y}_{1:2}) = \frac{\gamma_2(\boldsymbol{x}_{1:2})}{q_2(\boldsymbol{x}_{1:2} \mid \boldsymbol{y}_{1:2})} = \frac{p(\boldsymbol{x}_{1:2}, \boldsymbol{y}_{1:2})}{q_1(x_1 \mid y_1)\, q_2(x_2 \mid \boldsymbol{y}_{1:2}, x_1)} =$$

$$= \frac{p(x_1, y_1)}{q_1(x_1 \mid y_1)} \frac{p(\boldsymbol{x}_{1:2}, \boldsymbol{y}_{1:2})}{p(x_1, y_1)\, q_2(x_2 \mid \boldsymbol{y}_{1:2}, x_1)} = \underbrace{w_1(x_1, y_1)}_{\text{Weight from step 1}} \underbrace{\frac{p(\boldsymbol{x}_{1:2}, \boldsymbol{y}_{1:2})}{p(x_1, y_1)\, q_2(x_2 \mid \boldsymbol{y}_{1:2}, x_1)}}_{\text{Incremental weight}}$$

Sequential Importance Sampling

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

$$W_2^{(i)} \propto w_2\left(\boldsymbol{x}_{1:2}, \boldsymbol{y}_{1:2}\right) = \underbrace{w_1\left(x_1, y_1\right)}_{\text{Weight from step 1}} \underbrace{\frac{p\left(\boldsymbol{x}_{1:2}, \boldsymbol{y}_{1:2}\right)}{p\left(x_1, y_1\right) q_2\left(x_2 \mid \boldsymbol{y}_{1:2}, x_1\right)}}_{\text{Incremental weight}}$$

- Generalizing to step n, we can write:

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

$$= q_1\left(x_1 \mid y_1\right) \prod_{k=2}^{n} q_k\left(x_k \mid \boldsymbol{y}_{1:k}, \boldsymbol{x}_{1:k-1}\right)$$

- Thus if

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

we sample X_n from

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

Sequential Importance Sampling

- The weights for step n are then given as:

$$w_n(\boldsymbol{X}_{1:n}^{(i)},\boldsymbol{y}_{1:n})=\frac{p(\boldsymbol{X}_{1:n}^{(i)},\boldsymbol{y}_{1:n})}{q_n(\boldsymbol{X}_{1:n}^{(i)}\mid\boldsymbol{y}_{1:n})}=\underbrace{\frac{p(\boldsymbol{X}_{1:n-1}^{(i)},\boldsymbol{y}_{1:n-1})}{q_{n-1}(\boldsymbol{X}_{1:n-1}^{(i)}\mid\boldsymbol{y}_{1:n-1})}}_{w_{n-1}(\boldsymbol{X}_{1:n-1}^{(i)},\boldsymbol{y}_{1:n-1})}\frac{p(\boldsymbol{X}_{1:n}^{(i)},\boldsymbol{y}_{1:n})}{p(\boldsymbol{X}_{1:n-1}^{(i)},\boldsymbol{y}_{1:n-1})q_n(X_n^{(i)}\mid\boldsymbol{X}_{1:n-1}^{(i)},\boldsymbol{y}_{1:n})}$$

$$=w_{n-1}(\boldsymbol{X}_{1:n-1}^{(i)},\boldsymbol{y}_{1:n-1})\frac{p(\boldsymbol{X}_{1:n}^{(i)},\boldsymbol{y}_{1:n})}{p(\boldsymbol{X}_{1:n-1}^{(i)},\boldsymbol{y}_{1:n-1})q_n(X_n^{(i)}\mid\boldsymbol{X}_{1:n-1}^{(i)},\boldsymbol{y}_{1:n})}$$

- Similarly the normalized weights are as follows:

$$W_n^{(i)}\equiv W_n(\boldsymbol{X}_{1:n}^{(i)},\boldsymbol{y}_{1:n})\propto w_n(\boldsymbol{X}_{1:n}^{(i)},\boldsymbol{y}_{1:n})$$

- For our state space model, the above update formula takes the form:

$$w_n(\boldsymbol{X}_{1:n}^{(i)},\boldsymbol{y}_{1:n})=w_{n-1}(\boldsymbol{X}_{1:n-1}^{(i)},\boldsymbol{y}_{1:n-1})\frac{f(X_n^{(i)}\mid X_{n-1}^{(i)})g(y_n\mid X_n^{(i)})}{q_n(X_n^{(i)}\mid\boldsymbol{y}_{1:n},\boldsymbol{X}_{1:n-1}^{(i)})}$$

- In general, we may need to store all the paths $\left\{\boldsymbol{X}_{1:n}^{(i)}\right\}$ even if our interest is to only compute $\pi_n(x_n)=p\left(x_n\mid\boldsymbol{y}_{1:n}\right)$

Need for a Sequential Sampling Approach

- From practical perspective, we use proposal distributions of the form:

$$q_n\left(x_n \mid \boldsymbol{y}_{1:n}, \boldsymbol{x}_{1:n-1}\right) = q_n\left(x_n \mid y_n, x_{n-1}\right)$$

- The idea here is that given $\boldsymbol{x}_{n-1}$, $\boldsymbol{y}_{1:n-1}$ and $\boldsymbol{x}_{1:n-2}$ don’t bring any new information about X_n.

- Our sequential importance sampling update now looks as follows:

$$\underbrace{q_n\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)}_{Importance\ Samping\ at\ n} = \underbrace{q_{n-1}\left(\boldsymbol{x}_{1:n-1} \mid \boldsymbol{y}_{1:n-1}\right)}_{Distribution\ of\ the\ paths\ \boldsymbol{X}_{1:n-1}^{(i)}} \underbrace{q_n\left(x_n \mid y_n, x_{n-1}\right)}_{Conditional\ Distribution\ of\ X_n^{(i)}}$$

$$= q\left(x_1\right) \prod_{k=2}^{n} q_k\left(x_k \mid y_k, x_{k-1}\right)$$

- Thus we assume that at $n-1$ we have sampled $\boldsymbol{X}_{1:n-1}^{(i)} \sim q_{n-1}\left(\boldsymbol{x}_{1:n-1} \mid \boldsymbol{y}_{1:n-1}\right)$ and to obtain $\boldsymbol{X}_{1:n}^{(i)} \sim q\left(\boldsymbol{x}_{1:n} \mid \boldsymbol{y}_{1:n}\right)$, we need to sample $X_n^{(i)} \sim q_n\left(x_n \mid y_{n,} X_{n-1}^{(i)}\right)$ and then set

$$\boldsymbol{X}_{1:n}^{(i)} = \left(\underbrace{\boldsymbol{X}_{1:n-1}^{(i)}}_{Previously\ Sampled\ Paths}, \underbrace{X_n^{(i)}}_{Sampled\ Single\ Component\ at\ time\ n} \right)$$

Sequential Importance Sampling

- We now need to show that we can recursively compute estimates of our target distribution $p(\boldsymbol{x}_{1:n}\,|\,\boldsymbol{y}_{1:n})$ as well as of $p(\boldsymbol{y}_{1:n})$

- From our earlier Importance Sampling approximations:

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

$$\hat{p}_N(\boldsymbol{y}_{1:n}) = \frac{1}{N}\sum_{i=1}^{N} w\left(\boldsymbol{X}_{1:n}^{(i)},\boldsymbol{y}_{1:n}\right)$$

- We can show the following recursions for calculations of these weights:

$$w(\boldsymbol{x}_{1:n},\boldsymbol{y}_{1:n}) = \frac{p(\boldsymbol{x}_{1:n},\boldsymbol{y}_{1:n})}{q(\boldsymbol{x}_{1:n}\,|\,\boldsymbol{y}_{1:n})} = \underbrace{\frac{p(\boldsymbol{x}_{1:n-1},\boldsymbol{y}_{1:n-1})}{q(\boldsymbol{x}_{1:n-1}\,|\,\boldsymbol{y}_{1:n-1})}}_{w(\boldsymbol{x}_{1:n-1},\boldsymbol{y}_{1:n-1})}\underbrace{\frac{f(x_n\,|\,x_{n-1})\,g(y_n\,|\,x_n)}{q(x_n\,|\,y_{n,}x_{n-1})}}_{Incremental\ Weight}$$

- This suggests the following sequential Importance Sampling Algorithm.

Sequential Importance Sampling

At step $\boldsymbol{n} = \mathbf{1}$:

❑ Sample $X_1^{(i)} \sim q\left(x_1 \mid y_1\right), i = 1, \dots, N$ and then approximate:

$$\hat{p}_N(x_1|y_1) = \sum_{i=1}^{N} W_1^{(i)} \delta_{X_1^{(i)}}(x_1),\ W_1^{(i)}\left(X_1^{(i)}, y_1\right) \propto \frac{\mu\left(X_1^{(i)}\right) g\left(y_1, X_1^{(i)}\right)}{q\left(X_1^{(i)}|y_1\right)}$$

At step $\boldsymbol{n} \geq \mathbf{2}$:

❑ Sample $X_n^{(i)} \sim q\left(x_n \mid y_{n,} X_{n-1}^{(i)}\right), n = 1, \dots, N,$ and compute:

$$\hat{p}_N(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}) = \sum_{i=1}^{N} W_n^{(i)} \delta_{\boldsymbol{X}_{1:n}^{(i)}}(\boldsymbol{x}_{1:n}),$$

$$W_n^{(i)} \propto w\left(\boldsymbol{X}_{1:n}^{(i)}, \boldsymbol{y}_{1:n}\right) = w\left(\boldsymbol{X}_{1:n-1}^{(i)}, \boldsymbol{y}_{1:n-1}\right) \frac{f\left(X_n^{(i)}|X_{n-1}^{(i)}\right) g\left(y_n|X_n^{(i)}\right)}{q\left(X_n^{(i)}|y_{n,}X_{n-1}^{(i)}\right)}$$

❑ The algorithm has computational complexity $\mathcal{O}(N)$ independent of n.

Sequential Importance Sampling

- Note that the complexity of the algorithm does not increase with n.

- The algorithm is fully parallelizable.

- Also note that if our interest is on computing the marginal posterior,

  $\hat{p}_N(x_n|\boldsymbol{y}_{1:n})$ (posterior filtered density), then we only need to store $\boldsymbol{X}^{(i)}_{n-1:n}$ rather than all the $\boldsymbol{X}^{(i)}_{1:n}$ paths

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

$$W_n^{(i)} \propto w\left(\boldsymbol{X}^{(i)}_{1:n}, \boldsymbol{y}_{1:n}\right) = w\left(\boldsymbol{X}^{(i)}_{1:n-1}, \boldsymbol{y}_{1:n-1}\right) \frac{f\left(X_n^{(i)}|X_{n-1}^{(i)}\right) g\left(y_n|X_n^{(i)}\right)}{q\left(X_n^{(i)}|y_n, X_{n-1}^{(i)}\right)}$$

- One can show that this approaches the true posterior as $N \to \infty$.
  - Crisan, D., P. D. Moral, and T. Lyons (1999). Discrete filtering using branching and interacting particle systems. *Markov Processes and Related Fields* 5(3), 293–318.