Sequential Monte Carlo

Geir Storvik

Geilo Winter school 2023

UiO : Universitetet i Oslo

Norsk Regnesentral
NORWEGIAN COMPUTING CENTER

Outline

Sequential Monte Carlo

Static versus dynamic inference

- MCMC/INLA: Inference when all data is collected
- Assume now $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots$ are collected dynamically in time
- Want to do inference based on $\boldsymbol{y}_{1:t}$ at each time point t
- Can in principle start MCMC/INLA from scratch
- Possible to utlize compuation performed at time $t-1$?
  - YES: by sequential Monte Carlo

Sequential updating

- Aim now: Sequential sampling from $\pi_t(\boldsymbol{x}_t)$ for $t = 1, 2, \ldots$
- State space settings:

$$\begin{aligned} \boldsymbol{x}_t \sim & p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1};\boldsymbol{\theta}) && \text{State process} \\ \boldsymbol{y}_t \sim & p(\boldsymbol{y}_t|\boldsymbol{x}_t;\boldsymbol{\theta}) && \text{Observation process} \end{aligned}$$

  Aim: $\pi_t(\boldsymbol{x}_t) = p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ or $\pi_t(\boldsymbol{x}_{1:t}) = p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t})$
- Complex Bayesian settings
  - $p(\boldsymbol{x}|\boldsymbol{y})$ complex, $p(\boldsymbol{x})$ simple
  - Construct $\pi_t(\boldsymbol{x}) \propto p(\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x})^{\gamma_t}$ with $\gamma_t = \frac{t}{T}, t = 0, \ldots, T$
- Sequential updating: Breaks down high-dimensional sampling to many low-dimensional ones
- References:
  - Dai et al. (2022): *An invitation to sequential Monte Carlo samplers*
  - Naesseth et al. (2019): *Elements of sequential Monte Carlo*
  - Doucet et al. (2001): *Sequential Monte Carlo methods in practice*
  - Chopin et al. (2020): *An introduction to sequential Monte Carlo*

Sequential Monte Carlo

- Assume $\mathbf{x} = \mathbf{x}_{1:t} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t)$ have a Markov structure

$$\pi_t(\mathbf{x}_{1:t}) = \pi_1(\boldsymbol{x}_1) \prod_{s=2}^{t} \pi_s(\boldsymbol{x}_s | \boldsymbol{x}_{s-1})$$

Sequential Monte Carlo

- Assume $\mathbf{x} = \mathbf{x}_{1:t} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t)$ have a Markov structure

$$\pi_t(\mathbf{x}_{1:t}) = \pi_1(\boldsymbol{x}_1) \prod_{s=2}^{t} \pi_s(\boldsymbol{x}_s | \boldsymbol{x}_{s-1})$$

- Also assume a proposal distribution with Markov property:

$$g_t(\mathbf{x}_{1:t}) = g_1(\boldsymbol{x}_1) \prod_{s=2}^{t} g_s(\boldsymbol{s}_i | \boldsymbol{x}_{s-1})$$

Sequential Monte Carlo

- Assume $\mathbf{x} = \mathbf{x}_{1:t} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t)$ have a Markov structure

$$\pi_t(\mathbf{x}_{1:t}) = \pi_1(\boldsymbol{x}_1) \prod_{s=2}^{t} \pi_s(\boldsymbol{x}_s | \boldsymbol{x}_{s-1})$$

- Also assume a proposal distribution with Markov property:

$$g_t(\mathbf{x}_{1:t}) = g_1(\boldsymbol{x}_1) \prod_{s=2}^{t} g_s(\boldsymbol{s}_i | \boldsymbol{x}_{s-1})$$

- Importance weights:

$$w(\mathbf{x}_{1:t}) = \frac{\pi_t(\mathbf{x}_{1:t})}{g_t(\mathbf{x}_{1:t})} = \frac{\pi_1(\boldsymbol{x}_1)}{g_1(\boldsymbol{x}_1)} \prod_{s=2}^{t} \frac{\pi_s(\boldsymbol{x}_s | \boldsymbol{x}_{s-1})}{g_s(\boldsymbol{x}_s | \boldsymbol{x}_{s-1}))} = w(\boldsymbol{x}_{1:t-1}) \frac{\pi_t(\boldsymbol{x}_t | \boldsymbol{x}_{t-1})}{g_t(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}))}$$

Sequential Monte Carlo

- Assume $\mathbf{x} = \mathbf{x}_{1:t} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_t)$ have a Markov structure

$$\pi_t(\mathbf{x}_{1:t}) = \pi_1(\boldsymbol{x}_1)\prod_{s=2}^{t}\pi_s(\boldsymbol{x}_s|\boldsymbol{x}_{s-1})$$

- Also assume a proposal distribution with Markov property:

$$g_t(\mathbf{x}_{1:t}) = g_1(\boldsymbol{x}_1)\prod_{s=2}^{t}g_s(\boldsymbol{s}_i|\boldsymbol{x}_{s-1})$$

- Importance weights:

$$w(\mathbf{x}_{1:t}) = \frac{\pi_t(\mathbf{x}_{1:t})}{g_t(\mathbf{x}_{1:t})} = \frac{\pi_1(\boldsymbol{x}_1)}{g_1(\boldsymbol{x}_1)}\prod_{s=2}^{t}\frac{\pi_s(\boldsymbol{x}_s|\boldsymbol{x}_{s-1})}{g_s(\boldsymbol{x}_s|\boldsymbol{x}_{s-1}))} = w(\boldsymbol{x}_{1:t-1})\frac{\pi_t(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}{g_t(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}))}$$

- Opens up for sequential sampling/estimation
- Note: Possible to generalize to non-Markov settings as well
    - More computing at each step

Sequential Monte Carlo

Algorithm 1 SMC

1: Sample $\boldsymbol{x}_1 \sim g_1(\cdot)$. Let $w_1 = u_1 = \pi_1(\boldsymbol{x}_1)/g_1(\boldsymbol{x}_1)$. Set $t = 2$

2: Sample $\boldsymbol{x}_t|\boldsymbol{x}_{t-1} \sim g_t(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$.

3: Append $\boldsymbol{x}_t$ to $\boldsymbol{x}_{1:t-1}$, obtaining $\boldsymbol{x}_t$

4: Let $u_t = \pi_t(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})/g_t(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$

5: Let $w_t = w_{t-1}u_t$, the importance weight for $\boldsymbol{x}_{1:t}$

6: Increment t and return to step 2

Sequential Monte Carlo

Algorithm 2 SMC

1: Sample $\boldsymbol{x}_1 \sim g_1(\cdot)$. Let $w_1 = u_1 = \pi_1(\boldsymbol{x}_1)/g_1(\boldsymbol{x}_1)$. Set $t = 2$
2: Sample $\boldsymbol{x}_t|\boldsymbol{x}_{t-1} \sim g_t(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$.
3: Append $\boldsymbol{x}_t$ to $\boldsymbol{x}_{1:t-1}$, obtaining $\boldsymbol{x}_t$
4: Let $u_t = \pi_t(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})/g_t(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$
5: Let $w_t = w_{t-1}u_t$, the importance weight for $\boldsymbol{x}_{1:t}$
6: Increment t and return to step 2

- Can simulate N sequences in parallel!
- Approximation: $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) \approx \sum_{i=1}^N w_t^i \delta_{\boldsymbol{x}_t^i}(\boldsymbol{x}_t)$
  - Typically one would normalize weights

$$w_t^i \quad \rightarrow \quad w_t^i / \sum_j w_t^j$$

Weight degeneracy

- General rule:

$$\text{var}[Y] = E[\text{var}[Y|Z]] + \text{var}[E[Y|Z]] \geq \text{var}[E[Y|Z]]$$

Weight degeneracy

- General rule:

$$\text{var}[Y] = E[\text{var}[Y|Z]] + \text{var}[E[Y|Z]] \geq \text{var}[E[Y|Z]]$$

- $Y = w_t, Z = \boldsymbol{x}_{1:t-1}$ (w_{t-1} given by $\boldsymbol{x}_{1:t-1}$):

$$\begin{aligned} E[w_t|\boldsymbol{x}_{1:t-1}] =& w_{t-1} E[\tfrac{\pi_t(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}{g_t(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}|\boldsymbol{x}_{1:t-1}] \\ =& w_{t-1} \cdot 1 = w_{t-1} \end{aligned}$$

Weight degeneracy

- General rule:

$$\text{var}[Y] = E[\text{var}[Y|Z]] + \text{var}[E[Y|Z]] \geq \text{var}[E[Y|Z]]$$

- $Y = w_t, Z = \boldsymbol{x}_{1:t-1}$ (w_{t-1} given by $\boldsymbol{x}_{1:t-1}$):

$$\begin{aligned} E[w_t|\boldsymbol{x}_{1:t-1}] =& w_{t-1} E[\frac{\pi_t(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}{g_t(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}|\boldsymbol{x}_{1:t-1}] \\ =& w_{t-1} \cdot 1 = w_{t-1} \end{aligned}$$

implying that

$$\text{var}[w_t] \geq \text{var}[w_{t-1}]$$

which indicates that the variance will increase at each time-step.

Weight degeneracy

- General rule:

$$\text{var}[Y] = E[\text{var}[Y|Z]] + \text{var}[E[Y|Z]] \geq \text{var}[E[Y|Z]]$$

- $Y = w_t, Z = \mathbf{x}_{1:t-1}$ (w_{t-1} given by $\mathbf{x}_{1:t-1}$):

$$\begin{aligned} E[w_t|\mathbf{x}_{1:t-1}] =& w_{t-1} E[\tfrac{\pi_t(\mathbf{x}_t|\mathbf{x}_{t-1})}{g_t(\mathbf{x}_t|\mathbf{x}_{t-1})}|\mathbf{x}_{1:t-1}] \\ =& w_{t-1} \cdot 1 = w_{t-1} \end{aligned}$$

  implying that

$$\text{var}[w_t] \geq \text{var}[w_{t-1}]$$

  which indicates that the variance will increase at each time-step.
- Practical consequence:
  - Only a few samples will dominate the others
  - Variability of estimate will increase

Bootstrap filter for state space models

- Introduce resampling
  - Discard samples (particles) with small weight
  - Duplicate particles with high weight

Algorithm 3 SMC

1: Simulate $\boldsymbol{x}_1^i \sim p(\boldsymbol{x}_1)$ for $i = 1, \ldots, N$. ▷ Initialization
2: Put weights $w_1^i = p(\boldsymbol{y}_1|\boldsymbol{x}_1^i)$.
3: Sample $\{B_1^1, \ldots, B_1^N\}$ from $\{1, \ldots, N\}$ with probabilities $\{w_1^i\}$.
4: **for** $t = 2, 3, \ldots$ **do** ▷ Sequential Monte Carlo
5: Simulate $\boldsymbol{x}_t^i \sim p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{B_{t-1}^i})$ for $i = 1, \ldots, N$.
6: Put weights $w_t^i = p(\boldsymbol{y}_t|\boldsymbol{x}_t^i)$.
7: Sample $\{B_t^1, \ldots, B_t^N\}$ from $\{1, \ldots, N\}$ with probabilities $\{w_t^i\}$.
8: **end for**

- Approximation: $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) \approx \sum_{i=1}^N w_t^i \delta_{\boldsymbol{x}_t^i}(\boldsymbol{x}_t)$
- More general: Optional resampling with weights propagated if no resampling

State space models

- Model

$$\begin{aligned} \boldsymbol{x}_t &\sim p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}; \boldsymbol{\theta}) && \text{State process} \\ \boldsymbol{y}_t &\sim p(\boldsymbol{y}_t | \boldsymbol{x}_t; \boldsymbol{\theta}) && \text{Observation process} \end{aligned}$$

- Target distributions (assuming for now $\boldsymbol{\theta}$ known):

$$\pi(\boldsymbol{x}_{1:t}) = p(\boldsymbol{x}_{1:t} | \boldsymbol{y}_{1:t}) \propto p(\boldsymbol{x}_1) p(\boldsymbol{y}_1 | \boldsymbol{x}_1) \prod_{s=2}^{t} p(\boldsymbol{x}_s | \boldsymbol{x}_{s-1}) p(\boldsymbol{y}_s | \boldsymbol{x}_s)$$

- Unknown normalization constant(s):

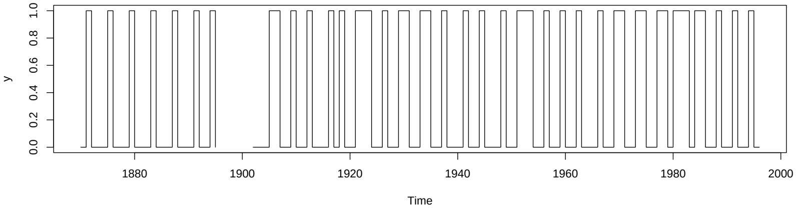
$$p(\boldsymbol{x}_t | \boldsymbol{y}_{1:t}) \approx \sum_{i=1}^{N} W_t^i \delta_{\boldsymbol{x}_t^i}(\boldsymbol{x}_t)$$

with $W_t^i = \frac{w_t^i}{\sum_j w_t^j}$

Lemmings data

- Observations: $y_t \in \{0, 1\}$, =1 if "lemming year"
- Possible simple model: $\boldsymbol{x}_t = \log(N_t)$

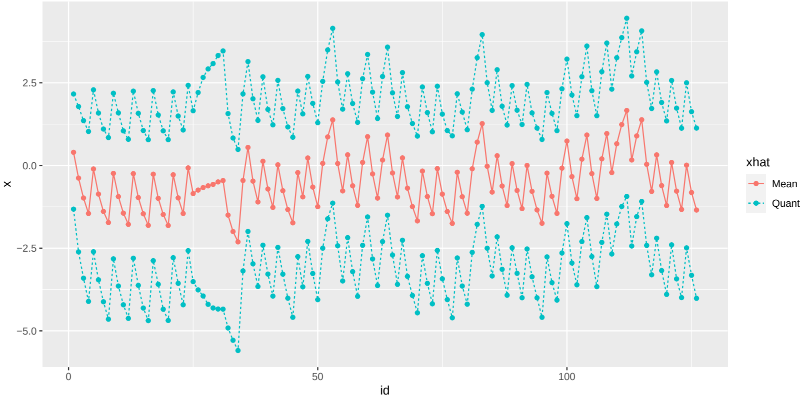
$$x_t = ax_{t-1} + x_{t-2} + \sigma\varepsilon_t \qquad\qquad \varepsilon_t \sim N(0, 1)$$



- Of interest: $p(N_t|y_{1:t})$

Lemmings - results

Script: `SMC_lemmings.R`



The many uses of Markov

- Markov chain Monte Carlo
- Markov assumption in Ising model:

$$p(x_i|\boldsymbol{x}_{-i}) = p(x_i|\boldsymbol{x}_{N_i})$$

- SMC: Markov structure in $\pi_t(\boldsymbol{x}_{1:t}) = \pi_1(\boldsymbol{x}_1)\prod_{s=2}^{t}\pi_s(\boldsymbol{x}_s|\boldsymbol{x}_{s-1})$
- SMC: Markov structure in $g_t(\boldsymbol{x}_{1:t}) = g_1(\boldsymbol{x}_1)\prod_{s=2}^{t}g_s(\boldsymbol{s}_i|\boldsymbol{x}_{s-1})$

Discuss the different uses of Markov assumptions

Details on SMC

Effective sample size

- Assume $w_i = w(\boldsymbol{x}_i), i = 1, ..., N$ are normalized weights
- Define effective sample size by

$$
\begin{aligned}
\widehat{N}_{eff} =& \frac{1}{\sum_{i=1}^{N} w_i^2} \\
=& N && \text{if } w_i = \tfrac{1}{n} \text{ for all } i \\
=& N - z && \text{if } w_i = 0, i \leq z, w_i = \tfrac{1}{N-z}, i > z \\
=& 1 && \text{if } w_j = 1, w_i = 0, i \neq j
\end{aligned}
$$

- General: Resampling introduce extra Monte Carlo variability
- Rule of tump: Resample only if $\widehat{N}_{eff} < 0.5N$

Resampling

- Simplest option:
  - Resample with probabilities equal to w_t^i.
  - Put weights on resample to $\tilde{w}_t^i = N^{-1}$
  - Number of repeats of $\boldsymbol{x}_t^i$, N_t^i is Binomial(N, w_t^i)
  - $E[N_t^i \tilde{w}_t^i] = N w_t^i$

Resampling

- Simplest option:
  - Resample with probabilities equal to w_t^i.
  - Put weights on resample to $\tilde{w}_t^i = N^{-1}$
  - Number of repeats of $\boldsymbol{x}_t^i$, N_t^i is Binomial(N, w_t^i)
  - $E[N_t^i \tilde{w}_t^i] = N w_t^i$
- More general resampling strategies are possible
- Sufficient requirement: $E[N_t^i \tilde{w}_t^i] = N w_t^i$

Resampling

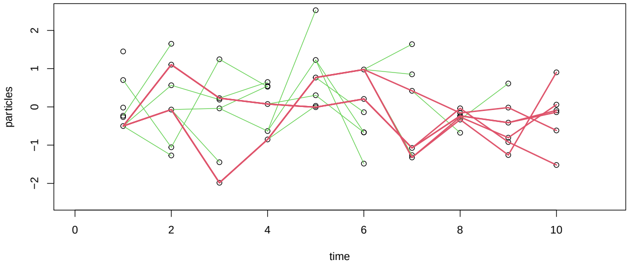
- Simplest option:
  - Resample with probabilities equal to w_t^i.
  - Put weights on resample to $\tilde{w}_t^i = N^{-1}$
  - Number of repeats of $\boldsymbol{x}_t^i$, N_t^i is Binomial(N, w_t^i)
  - $E[N_t^i \tilde{w}_t^i] = Nw_t^i$
- More general resampling strategies are possible
- Sufficient requirement: $E[N_t^i \tilde{w}_t^i] = Nw_t^i$
- Optimal strategy (for equally weighted samples)
  - For $i = 1, ..., N$, put ($\lfloor a \rfloor$ is the largest integer smaller than a)

    $$\widetilde{N}_t^i = \lfloor Nw_t^i \rfloor \quad \text{(Some will be zero)}$$

  - Let $\delta_t^i = w_t^i - \widetilde{N}_t^i / N$
  - Define $K = N - \sum_{i=1}^{N} \widetilde{N}_t^i$ (remaining particles that have not been allocated)
  - Sample $(D_t^1, ..., D_t^N)$ from the multinomial distribution with probabilities proportional to $(\delta_t^1, ..., \delta_t^N)$.
  - Put $N_t^i = \tilde{N}_t^i + D_t^i$
  - Make N_t^i replicates of $\boldsymbol{x}_t^i$, but all weights to $1/N$

Resampling - degeneracy

- At time t: Samples $\{\boldsymbol{x}_{1:t}^i, i = 1, ..., N\}$ from $p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t})$
- When resampling, resample whole vector $\boldsymbol{x}_{1:t}$
- When repeated resampling at many time-steps, $\boldsymbol{x}_1$ is resampled each time, less and less unique values



Why did it work in the Lemmings example?

- The results based on $p(x_t|\boldsymbol{y}_{1:t})$
- Degeneracy problem related to $p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t})$
  - but would be a problem also for $p(x_t|\boldsymbol{y}_{1:t})$ if no resampling!
- Theoretical properties:
  - If interest in $p(x_t|\boldsymbol{y}_{1:t})$: Error will be uniform over time
  - If interest in $p(x_{1:t}|\boldsymbol{y}_{1:t})$: Error will increase (exponential) over time

Marginal likelihood

- We have

$$L(\boldsymbol{\theta}) = p(\boldsymbol{y}_{1:T}|\boldsymbol{\theta}) = p(\boldsymbol{y}_1|\boldsymbol{\theta}) \prod_{t=2}^{T} p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1};\boldsymbol{\theta})$$

- Estimate of $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})$:

$$\hat{p}(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}) = \frac{1}{N}\sum_{i=1}^{N} w_t^i$$

- Estimate of marginal likelihood:

$$\hat{L}(\boldsymbol{\theta}) = \prod_{t=1}^{T}\left(\frac{1}{N}\sum_{i=1}^{N} w_t^i\right)$$

- Can show that $\hat{L}(\boldsymbol{\theta})$ is an unbiased estimator of $L(\boldsymbol{\theta})$

Feynman-Kac formulation

Reformulation of target density

- State space models:

$$p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t}) \propto p(x_1)p(y_1|x_1)\prod_{s=2}^{t} p(x_s|x_{s-1})p(y_s|x_s)$$

- Using that $p(x_s|x_{s-1})p(y_s|x_s) = p(x_s, y_s|x_{s-1}) = p(x_s|x_{s-1}, y_s)p(y_s|x_{s-1})$, we have

$$p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t}) \propto p(x_1|y_1)p(y_1)\prod_{s=2}^{t} p(x_s|x_{s-1}, y_s)p(y_s|x_{s-1})$$

- Indicate different sampling strategy:
  - Simulate $x_s \sim p(x_s|x_{s-1}, y_s)$
  - Update weights with $u_t = p(y_s|x_{s-1})$
- Will typically give better proposals
  - But more difficulty in calculating weights

Algorithm general

Algorithm 4 Guided Particle filter

1: Simulate $\boldsymbol{x}_1^i \sim q(\boldsymbol{x}_1)$ for $i = 1, ..., N$. ▷ Initialization
2: Put weights $w_1^i = p(\boldsymbol{y}_1|\boldsymbol{x}_1^i)\frac{p(\boldsymbol{x}_1)}{q(\boldsymbol{x}_1^i)}$.
3: Sample $\{B_1^1, ..., B_1^N\}$ from $\{1, ..., N\}$ with probabilities $\{w_1^i\}$.
4: Put $w_1^i = 1/N$.
5: **for** $t = 2, 3, ...$ **do** ▷ Sequential Monte Carlo
6: Simulate $\boldsymbol{x}_t^i \sim q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{B_{t-1}^i}, \boldsymbol{y}_t)$ for $i = 1, ..., N$.
7: Put weights $w_t^i = p(\boldsymbol{y}_t|\boldsymbol{x}_t^i)\frac{p(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^{B_{t-1}^i})}{q(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^{B_{t-1}^i}, \boldsymbol{y}_t)}$.
8: Sample $\{B_t^1, ..., B_t^N\}$ from $\{1, ..., N\}$ with probabilities $\{w_t^i\}$.
9: Put $w_t^i = 1/N$.
10: **end for**

- Approximation: $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) \approx \sum_{i=1}^N w_t^i \delta_{\boldsymbol{x}_t^i}(\boldsymbol{x}_t)$
- More general: Optimal resampling with weights propagated if no resampling

Feynman-Kac formulation - Chopin et al. (2020)

- Assume a general set of target distributions

$$Q_t(x_{1:t}) = \frac{1}{L_t} G_1(x_1) \left\{ \prod_{s=2}^{t} G_s(x_{s-1}, x_s) \right\} M_t(x_{1:t})$$

$$M_t(x_{1:t}) = M_1(x_1) \prod_{s=2}^{t} M_s(x_{s-1}, x_s)$$

Markov process

- Ordinary state space model:

$$M_s(x_{s-1}, x_s) = p(x_s | x_{s-1}) \qquad G_s(x_{s-1}, x_s) = p(y_s | x_s)$$

Feynman-Kac formulation - Chopin et al. (2020)

- Assume a general set of target distributions

$$Q_t(x_{1:t}) = \frac{1}{L_t} G_1(x_1) \left\{ \prod_{s=2}^{t} G_s(x_{s-1}, x_s) \right\} M_t(x_{1:t})$$

$$M_t(x_{1:t}) = M_1(x_1) \prod_{s=2}^{t} M_s(x_{s-1}, x_s) \qquad \text{Markov process}$$

- Ordinary state space model:

$$M_s(x_{s-1}, x_s) = p(x_s|x_{s-1}) \qquad G_s(x_{s-1}, x_s) = p(y_s|x_s)$$

- Reformulated model

$$M_s(, x_{s-1} x_s) = p(x_s|x_{s-1}, y_s) \qquad G_s(x_{s-1}, x_s) = p(y_s|x_{s-1})$$

- Possible with other reformulations as long as

$$G_s(x_{s-1}, x_s) M_s(x_{s-1}, x_s) = p(y_s|x_s) p(x_s|x_{s-1})$$

Why Feynman-Kac formalism?

- Different formulations share the same fundamental structure
  - Can be exploited for obtaining theoretical results
  - Can construct/exploit a variety of SMC algorithms in a common framework
- Ideal for development of generic software
  - Chopin et al. (2020): `particles` library (python)
  - Different algorithms correspond to "Bootstrap filter" for reformulated models
    - Bootstrap filter: Use $M_s(x_{s-1}, x_s)$ as proposal, use $G_s(x_{s-1}, x_s)$ as weight update.

Smoothing algorithms

Smoothing algorithms

- Algorithms so far target $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ or $p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t})$ - filtering
- In many cases interest in $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:T})$ or $p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:T})$
  - State space models (parameters known)

$$
\begin{array}{ccccccccc}
\longrightarrow & \boldsymbol{x}_{t-1} & \longrightarrow & \boldsymbol{x}_t & \longrightarrow & \boldsymbol{x}_{t+1} & \longrightarrow & \qquad & \text{Process} \\
 & \downarrow & & \downarrow & & \downarrow & & & \\
 & \boldsymbol{y}_{t-1} & & \boldsymbol{y}_t & & \boldsymbol{y}_{t+1} & & & \text{Observations}
\end{array}
$$

Smoothing algorithms

- Algorithms so far target $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ or $p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t})$ - filtering
- In many cases interest in $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:T})$ or $p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:T})$
  - State space models (parameters known)

$$
\begin{array}{ccccccc}
\longrightarrow & \boldsymbol{x}_{t-1} & \longrightarrow & \boldsymbol{x}_t & \longrightarrow & \boldsymbol{x}_{t+1} & \longrightarrow \\
 & \downarrow & & \downarrow & & \downarrow & \\
 & \boldsymbol{y}_{t-1} & & \boldsymbol{y}_t & & \boldsymbol{y}_{t+1} &
\end{array}
$$

Process

Observations

  - Smoothing distributions

$$
\begin{aligned}
p(\boldsymbol{x}_{1:T}|\boldsymbol{y}_{1:T}) =& p(\boldsymbol{x}_T|\boldsymbol{y}_{1:T}) \prod_{t=T-1}^{1} p(\boldsymbol{x}_t|\boldsymbol{x}_{t+1:T}, \boldsymbol{y}_{1:T}) \\
=& p(\boldsymbol{x}_T|\boldsymbol{y}_{1:T}) \prod_{t=T-1}^{1} p(\boldsymbol{x}_t|\boldsymbol{x}_{t+1}, \boldsymbol{y}_{1:t})
\end{aligned}
$$

Further

$$
p(\boldsymbol{x}_t|\boldsymbol{x}_{t+1}, \boldsymbol{y}_{1:t}) = \frac{p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})}{p(\boldsymbol{x}_{t+1}|\boldsymbol{y}_{1:t})}
$$

Smoothing algorithms - cont

$$p(\boldsymbol{x}_t|\boldsymbol{x}_{t+1:T},\boldsymbol{y}_{1:T}) = p(\boldsymbol{x}_t|\boldsymbol{x}_{t+1},\boldsymbol{y}_{1:t}) = \frac{p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})}{p(\boldsymbol{x}_{t+1}|\boldsymbol{y}_{1:t})}$$

- From filter algorithm:

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) \approx \sum_{i=1}^N w_t^i \delta_{\boldsymbol{x}_t^i}(\boldsymbol{x}_t)$$

$$p(\boldsymbol{x}_{t+1}|\boldsymbol{y}_{1:t}) = \int_{\boldsymbol{x}_t} p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})d\boldsymbol{x} \approx \sum_{i=1}^N w_t^i p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t^i)$$

- Combined:

$$\begin{aligned} p(\boldsymbol{x}_t|\boldsymbol{x}_{t+1},\boldsymbol{y}_{1:t}) &\approx \frac{\sum_{i=1}^N w_t^i p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t^i)\delta_{\boldsymbol{x}_t^i}(\boldsymbol{x}_t)}{\sum_{i=1}^N w_t^i p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t^i)} \\ &= \sum_{i=1}^N \widetilde{w}_t^i p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t^i)\delta_{\boldsymbol{x}_t^i}(\boldsymbol{x}_t) \qquad \widetilde{w}_t^i = w_t^i p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t^i) \end{aligned}$$

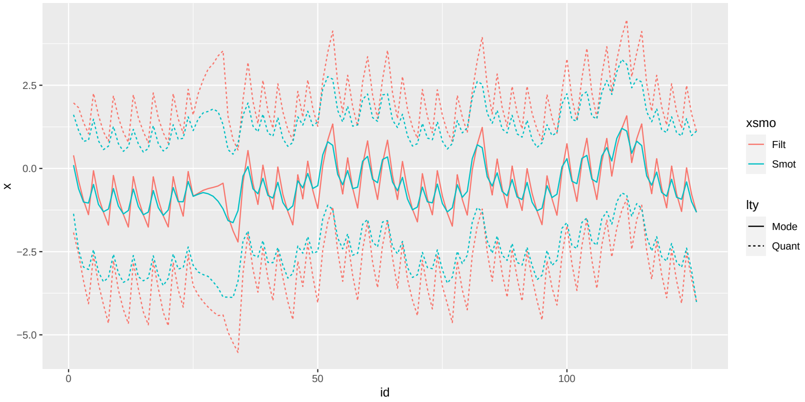
Smoothing algorithm

Algorithm 5 Particle smoother

1: Put $\widetilde{w}_T^i = w_T^i, i = 1, ..., N$
2: Sample $\{B_T^1, ..., B_T^N\}$ from $\{1, ..., N\}$ with probabilities $\{\widetilde{w}_T^i\}$. ▷ Initialization
3: **for** $t = T - 1, T - 2, ..., 1$ **do** ▷ Backwards smoothing
4: Calculate $\widetilde{w}_t^i = w_t^i \cdot p(\boldsymbol{x}_{t+1}^{B_{t+1}^i} | \boldsymbol{x}_t^i)$ for $i = 1, ..., N$.
5: Sample $\{B_T^1, ..., B_T^N\}$ from $\{1, ..., N\}$ with probabilities $\{\widetilde{w}_t^i\}$.
6: **end for**

- Approximation: $p(\boldsymbol{x}_t | \boldsymbol{y}_{1:T}) \approx \sum_{i=1}^N \widetilde{w}_t^i \delta_{\boldsymbol{x}_t^i}(\boldsymbol{x}_t)$
- Several other smoother algorithms

Lemmings - smoothing results



A Case study - Covid-19

Case study - Covid-19

See separate file

SMC and parameter estimation

SMC and parameter estimation

- Assume

$$
\begin{aligned}
\boldsymbol{x}_t &\sim p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}; \boldsymbol{\theta}) \\
y_t &\sim p(\boldsymbol{y}_t | \boldsymbol{x}_t; \boldsymbol{\theta})
\end{aligned}
$$

SMC and parameter estimation

- Assume

$$\begin{aligned} \boldsymbol{x}_t \sim& p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1};\boldsymbol{\theta}) \\ y_t \sim& p(\boldsymbol{y}_t|\boldsymbol{x}_t;\boldsymbol{\theta}) \end{aligned}$$

- Aim now: Simultaneous inference on $\boldsymbol{\theta}$
- Two main approaches:
  - Maximum likelihood: $\hat{\boldsymbol{\theta}}_{ML} = \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$
  - Bayesian approach: $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T}) \propto p(\boldsymbol{\theta})p(\boldsymbol{y}_{1:T}|\boldsymbol{\theta}) = p(\boldsymbol{\theta})L(\boldsymbol{\theta})$
- Important property of SMC: Unbiased estimate of marginal likelihood $L_t(\boldsymbol{\theta}) = p(\boldsymbol{y}_{1:T}|\boldsymbol{\theta})$:

$$\hat{L}_T(\boldsymbol{\theta}) = \prod_{t=1}^{T}\left(\frac{1}{N}\sum_{i=1}^{N} w_t^i\right)$$

- Two main classes of methods
  - Offline methods
  - Online methods

SMC and Bayesian parameter estimation

- Assume

$$
\begin{aligned}
\boldsymbol{x}_1 \sim & p(\boldsymbol{x}_1; \theta) \\
\boldsymbol{x}_t \sim & p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}; \theta) \\
Y_t \sim & p(\boldsymbol{y}_t | \boldsymbol{x}_t; \theta) \\
\theta \sim & p(\theta)
\end{aligned}
$$

- Aim now: Simulate from $p(\boldsymbol{x}_t, \theta | \boldsymbol{y}_{1:t})$
- Several approaches
  - Direct use of SMC
  - Introducing dynamics in θ
  - Using sufficient statistics
  - Particle MCMC

SMC and maximum likelihood

- Interested in maximizing

$$L_t(\theta) = p(\boldsymbol{y}_{1:t}|\theta) = \int_{\boldsymbol{x}_{1:t}} p(\boldsymbol{y}_{1:t}|\boldsymbol{x}_{1:t};\theta)p(\boldsymbol{x}_{1:t}|\theta)d\boldsymbol{x}_{1:t}.$$

SMC and maximum likelihood

- Interested in maximizing

$$L_t(\theta) = p(\boldsymbol{y}_{1:t}|\theta) = \int_{\boldsymbol{x}_{1:t}} p(\boldsymbol{y}_{1:t}|\boldsymbol{x}_{1:t};\theta)p(\boldsymbol{x}_{1:t}|\theta)d\boldsymbol{x}_{1:t}.$$

- Main problem: Calculation of the likelihood function (and possibly the score function in order to do optimization)

SMC and maximum likelihood

- Interested in maximizing

$$L_t(\theta) = p(\boldsymbol{y}_{1:t}|\theta) = \int_{\boldsymbol{x}_{1:t}} p(\boldsymbol{y}_{1:t}|\boldsymbol{x}_{1:t};\theta)p(\boldsymbol{x}_{1:t}|\theta)d\boldsymbol{x}_{1:t}.$$

- Main problem: Calculation of the likelihood function (and possibly the score function in order to do optimization)
- Main approach: Use that

$$L(\theta) = p(\boldsymbol{y}_{1:T}|\theta) = p(\boldsymbol{y}_1|\theta)\prod_{t=2}^{T} p(\boldsymbol{y}_s|\boldsymbol{y}_{1:s-1};\theta) \approx \prod_{t=1}^{T}\left(\frac{1}{N}\sum_{i=1}^{N} w_t^i\right)$$

- Poyiadjis et al. (2011): Algorithms for calculating the score function and information (matrix) recursively
- Can be used for gradient descent methods

Particle MCMC

- Andrieu et al. (2010)
- Ideal MCMC ($p(\theta|\mathbf{y}) \propto p(\theta)L(\theta)$):
  1. Sample $\theta^* \sim g(\theta^*|\theta)$
  2. Calculate M-H ratio $r = \frac{p(\theta^*)L(\theta^*)g(\theta|\theta^*)}{p(\theta)L(\theta)g(\theta^*|\theta)}$
  3. Accept θ^* with prob $\min\{1, r\}$

Particle MCMC

- Andrieu et al. (2010)
- Ideal MCMC ($p(\boldsymbol{\theta}|\boldsymbol{y}) \propto p(\boldsymbol{\theta})L(\boldsymbol{\theta})$):
  1. Sample $\boldsymbol{\theta}^* \sim g(\boldsymbol{\theta}^*|\boldsymbol{\theta})$
  2. Calculate M-H ratio $r = \frac{p(\boldsymbol{\theta}^*)L(\boldsymbol{\theta}^*)g(\boldsymbol{\theta}|\boldsymbol{\theta}^*)}{p(\boldsymbol{\theta})L(\boldsymbol{\theta})g(\boldsymbol{\theta}^*|\boldsymbol{\theta})}$
  3. Accept $\boldsymbol{\theta}^*$ with prob $\min\{1, r\}$
- Pseudo-Marginal algorithm:
  1. Sample $\boldsymbol{\theta}^* \sim g(\boldsymbol{\theta}^*|\boldsymbol{\theta})$
  2. Calculate $\hat{L}(\boldsymbol{\theta}^*)$
  3. Calculate M-H ratio $\hat{r} = \frac{\pi(\boldsymbol{\theta}^*)p(\boldsymbol{\theta}|\boldsymbol{\theta}^*)}{\pi(\boldsymbol{\theta})p(\boldsymbol{\theta}^*|\boldsymbol{\theta})}$
  4. Accept $\boldsymbol{\theta}^*$ with prob $\min\{1, \hat{r}\}$
- Particle MCMC: Use SMC to calculate $\hat{L}(\boldsymbol{\theta}^*)$

Direct use of SMC

- Assume at time $t-1$ the existence of a properly weighted sample $\{(\boldsymbol{x}_{t-1}^i, \theta^i, w_{t-1}^i)\}$ with respect to $p(\boldsymbol{x}_{t-1}, \theta | \boldsymbol{y}_{1:t-1})$.
- We have

$$\begin{aligned} p(\boldsymbol{x}_t, \theta | \boldsymbol{y}_{1:t-1}) &= \int_{\boldsymbol{x}_{t-1}} p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}, \theta) p(\boldsymbol{x}_{t-1}, \theta | \boldsymbol{y}_{1:t-1}) d\boldsymbol{x}_{t-1} \\ &\approx \sum_{i=1}^{N} w_{t-1}^i p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}^i, \theta^i) \delta_\theta(\theta^i) \end{aligned}$$

and

$$p(\boldsymbol{x}_t, \theta | \boldsymbol{y}_{1:t}) \approx c \cdot \sum_{i=1}^{N} w_{t-1}^i p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}^i, \theta^i) \delta_{\theta^i}(\theta) p(\boldsymbol{y}_t | \boldsymbol{x}_t, \theta^i)$$

- Updated samples $\{(\theta^i, \boldsymbol{x}_t^i, w_t^i)\}$:
  1. Simulate $\boldsymbol{x}_t^i \sim p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}^i, \theta^i)$
  2. Update the weights by $w_t^i \propto w_{t-1}^i p(\boldsymbol{y}_t | \boldsymbol{x}_t^i, \theta^i)$
- The sample $\{\theta^i\}$ do not change over time.
- With resampling, this will lead to degeneracy

Direct use of SMC - properly weighted?

- Proposal:

$$\theta^i \sim g(\theta)\boldsymbol{x}_s^i \sim p(\boldsymbol{x}_s|\boldsymbol{x}_{s-1}^i,\theta^i), \quad s = 1,...,t$$

- Weights at time $t = 1$:

$$w_1^i = \frac{p(\theta^i)p(\boldsymbol{x}_1^i|\theta^i)p(\boldsymbol{y}_1|\boldsymbol{x}_1^i,\theta^i)}{g(\theta)p(\boldsymbol{x}_1^i|\theta^i)} = \frac{p(\theta^i)p(\boldsymbol{y}_1|\boldsymbol{x}_1^i,\theta^i)}{g(\theta)}$$

giving properly weighted samples at time 1.

Direct use of SMC - properly weighted?

- Proposal:

$$\theta^i \sim g(\theta)\boldsymbol{x}_s^i \sim p(\boldsymbol{x}_s|\boldsymbol{x}_{s-1}^i,\theta^i), \quad s=1,...,t$$

- Weights at time $t=1$:

$$w_1^i = \frac{p(\theta^i)p(\boldsymbol{x}_1^i|\theta^i)p(\boldsymbol{y}_1|\boldsymbol{x}_1^i,\theta^i)}{g(\theta)p(\boldsymbol{x}_1^i|\theta^i)} = \frac{p(\theta^i)p(\boldsymbol{y}_1|\boldsymbol{x}_1^i,\theta^i)}{g(\theta)}$$

  giving properly weighted samples at time 1.
- At time t:

$$\begin{aligned}
w_t^i =& \frac{p(\theta^i)p(\boldsymbol{x}_1^i|\theta^i)p(\boldsymbol{y}_1|\boldsymbol{x}_1^i,\theta^i)\prod_{s=2}^t p(\boldsymbol{x}_s^i|\boldsymbol{x}_{s-1}^i,\theta^i)p(\boldsymbol{y}_s|\boldsymbol{x}_s^i,\theta^i)}{g(\theta)p(\boldsymbol{x}_1^i|\theta^i))\prod_{s=2}^t p(\boldsymbol{x}_s^i|\boldsymbol{x}_{s-1}^i,\theta^i)}\\
=& \frac{p(\theta^i)p(\boldsymbol{x}_1^i|\theta^i)p(\boldsymbol{y}_1|\boldsymbol{x}_1^i,\theta^i)\prod_{s=2}^t p(\boldsymbol{y}_s|\boldsymbol{x}_s^i,\theta^i)}{g(\theta)p(\boldsymbol{x}_1^i|\theta^i))}\\
=& \frac{p(\theta^i)p(\boldsymbol{x}_1^i|\theta^i)p(\boldsymbol{y}_1|\boldsymbol{x}_1^i,\theta^i)\prod_{s=2}^{t-1} p(\boldsymbol{y}_s|\boldsymbol{x}_s^i,\theta^i)}{g(\theta)p(\boldsymbol{x}_1^i|\theta^i))} p(\boldsymbol{y}_t|\boldsymbol{x}_t^u,\theta^i)\\
=& w_{t-1}^i p(\boldsymbol{y}_t|\boldsymbol{x}_t^u,\theta^i)
\end{aligned}$$

Direct use of SMC - properly weighted?

- Proposal:

$$\theta^i \sim g(\theta)\boldsymbol{x}_s^i \sim p(\boldsymbol{x}_s|\boldsymbol{x}_{s-1}^i, \theta^i), \quad s = 1, ..., t$$

- Weights at time $t = 1$:

$$w_1^i = \frac{p(\theta^i)p(\boldsymbol{x}_1^i|\theta^i)p(\boldsymbol{y}_1|\boldsymbol{x}_1^i, \theta^i)}{g(\theta)p(\boldsymbol{x}_1^i|\theta^i)} = \frac{p(\theta^i)p(\boldsymbol{y}_1|\boldsymbol{x}_1^i, \theta^i)}{g(\theta)}$$

  giving properly weighted samples at time 1.
- At time t:

$$\begin{aligned} w_t^i =& \frac{p(\theta^i)p(\boldsymbol{x}_1^i|\theta^i)p(\boldsymbol{y}_1|\boldsymbol{x}_1^i, \theta^i)\prod_{s=2}^t p(\boldsymbol{x}_s^i|\boldsymbol{x}_{s-1}^i, \theta^i)p(\boldsymbol{y}_s|\boldsymbol{x}_s^i, \theta^i)}{g(\theta)p(\boldsymbol{x}_1^i|\theta^i))\prod_{s=2}^t p(\boldsymbol{x}_s^i|\boldsymbol{x}_{s-1}^i, \theta^i)} \\ =& \frac{p(\theta^i)p(\boldsymbol{x}_1^i|\theta^i)p(\boldsymbol{y}_1|\boldsymbol{x}_1^i, \theta^i)\prod_{s=2}^t p(\boldsymbol{y}_s|\boldsymbol{x}_s^i, \theta^i)}{g(\theta)p(\boldsymbol{x}_1^i|\theta^i))} \\ =& \frac{p(\theta^i)p(\boldsymbol{x}_1^i|\theta^i)p(\boldsymbol{y}_1|\boldsymbol{x}_1^i, \theta^i)\prod_{s=2}^{t-1} p(\boldsymbol{y}_s|\boldsymbol{x}_s^i, \theta^i)}{g(\theta)p(\boldsymbol{x}_1^i|\theta^i))} p(\boldsymbol{y}_t|\boldsymbol{x}_t^u, \theta^i) \\ =& w_{t-1}^i p(\boldsymbol{y}_t|\boldsymbol{x}_t^u, \theta^i) \end{aligned}$$

- Main problem: Now we need to resample $(\theta, \boldsymbol{x}_{1:t})$.
  Will result in degeneracy when $p(\theta, \boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ is of interest.

Lemmings data

- Interested in the dynamics of the lemmings populations
- From church books: Binary records on lemmings years or not.

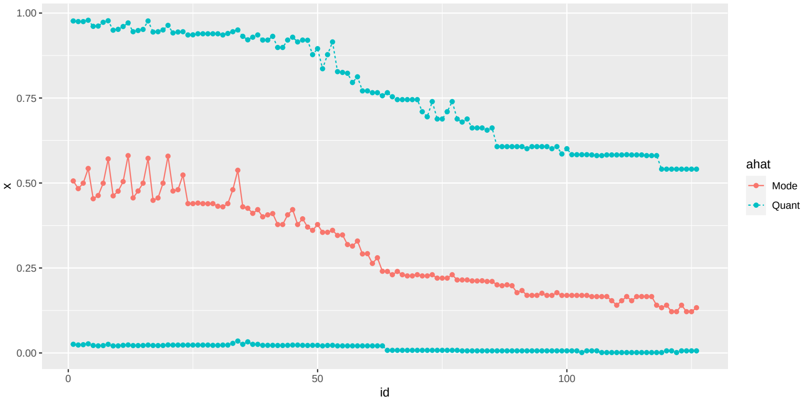
Lemmings data

- Interested in the dynamics of the lemmings populations
- From church books: Binary records on lemmings years or not.
- Define $\boldsymbol{x}_t = \log(N_t)$, N_t population size at year t
- Model

$$\begin{aligned}\boldsymbol{x}_t =& a\boldsymbol{x}_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim N(0, \sigma^2)\\ \boldsymbol{y}_t \sim& \text{Binom}\left(1, \tfrac{\exp(\boldsymbol{x}_t)}{1+\exp(\boldsymbol{x}_t)}\right)\end{aligned}$$

- Of interest: $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$, $p(a|\boldsymbol{y}_{1:t})$
- `SMC_lin_bin.R`, `SMC_lemmings_parest_direct.R`

Results - Lemmings



Introducing dynamics in θ

- Liu and West (2001): Assume θ is (slowly) changing with time:

$$\theta_t = \theta_{t-1} + \zeta_t, \quad \zeta_t \sim N(0, q)$$

- Focus on $p(\boldsymbol{x}_t, \theta_t | \boldsymbol{y}_{1:t})$.
- Assume a weighted sample $\{(\boldsymbol{x}_{t-1}^i, \theta_{t-1}^i, w_{t-1}^i)\}$

Introducing dynamics in θ

- Liu and West (2001): Assume θ is (slowly) changing with time:

$$\theta_t = \theta_{t-1} + \zeta_t, \quad \zeta_t \sim N(0, q)$$

- Focus on $p(\boldsymbol{x}_t, \theta_t | \boldsymbol{y}_{1:t})$.
- Assume a weighted sample $\{(\boldsymbol{x}_{t-1}^i, \theta_{t-1}^i, w_{t-1}^i)\}$

$$\begin{aligned} p(\boldsymbol{x}_t, \theta_t | \boldsymbol{y}_{1:t-1}) &= \int_{\boldsymbol{x}_{t-1}} p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}, \theta_t) p(\theta_t | \theta_{t-1}) p(\boldsymbol{x}_{t-1}, \theta_{t-1} | \boldsymbol{y}_{1:t-1}) d\boldsymbol{x}_{t-1} d\theta_{t-1} \\ &\approx \sum_{i=1}^N w_{t-1}^i p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}^i, \theta_t) p(\theta_t | \theta_{t-1}^i) \end{aligned}$$

$$p(\boldsymbol{x}_t, \theta_t | \boldsymbol{y}_{1:t}) \approx c \cdot \sum_{i=1}^N w_{t-1}^i p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}^i, \theta_t) p(\theta_t | \theta_{t-1}^i) p(\boldsymbol{y}_t | \boldsymbol{x}_t, \theta_t).$$

Introducing dynamics in θ

- Liu and West (2001): Assume θ is (slowly) changing with time:

$$\theta_t = \theta_{t-1} + \zeta_t, \quad \zeta_t \sim N(0, q)$$

- Focus on $p(\boldsymbol{x}_t, \theta_t | \boldsymbol{y}_{1:t})$.
- Assume a weighted sample $\{(\boldsymbol{x}_{t-1}^i, \theta_{t-1}^i, w_{t-1}^i)\}$

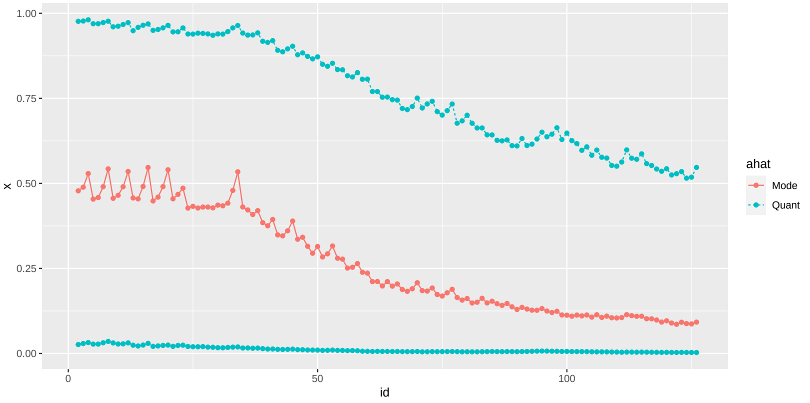
$$p(\boldsymbol{x}_t, \theta_t | \boldsymbol{y}_{1:t-1}) = \int_{\boldsymbol{x}_{t-1}} p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}, \theta_t) p(\theta_t | \theta_{t-1}) p(\boldsymbol{x}_{t-1}, \theta_{t-1} | \boldsymbol{y}_{1:t-1}) d\boldsymbol{x}_{t-1} d\theta_{t-1}$$

$$\approx \sum_{i=1}^{N} w_{t-1}^i p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}^i, \theta_t) p(\theta_t | \theta_{t-1}^i)$$

$$p(\boldsymbol{x}_t, \theta_t | \boldsymbol{y}_{1:t}) \approx c \cdot \sum_{i=1}^{N} w_{t-1}^i p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}^i, \theta_t) p(\theta_t | \theta_{t-1}^i) p(\boldsymbol{y}_t | \boldsymbol{x}_t, \theta_t).$$

- Update samples to $\{(\theta_t^i, \boldsymbol{x}_t^i, w_t^i)\}$ by
  1. Simulate $\theta_t^i \sim p(\theta_t | \theta_{t-1}^i)$,
  2. Simulate $\boldsymbol{x}_t^i \sim p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}^i, \theta_t^i)$
  3. Update the weights by $w_t^i \propto w_{t-1}^i p(\boldsymbol{y}_t | \boldsymbol{x}_t^i, \theta_t^i)$.
- `SMC_lemmings_parest_dyn.R`

Results - Lemmings



Dynamics in θ - continued

- New values $\{\theta_t^i\}$ are generated at each time point

Dynamics in θ - continued

- New values $\{\theta_t^i\}$ are generated at each time point
- Main problem: Introduce extra variability in θ_t.
- Consequence: Estimation of θ_t mainly based on most recent observations
- The model

$$\theta_t = \theta_{t-1} + \zeta_t, \quad \zeta_t \sim N(0, q)$$

  might be reasonable
- New problem: Estimate the static parameter q.
- `SMC_lin_bin_parest_dyn.R`

Sufficient statistics

- Example:

$$\boldsymbol{x}_t = a\boldsymbol{x}_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim N(0, \sigma^2), \quad \sigma \text{ known for simplicity}$$

- The distribution $p(\boldsymbol{y}_t|\boldsymbol{x}_t)$ can be arbitrary (but not depending on θ).
- $\theta = a$ needs to be estimated. Assume a prior $a \sim N(\mu_a, \sigma_a^2)$.

Sufficient statistics

- Example:

$$\mathbf{x}_t = a\mathbf{x}_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim N(0, \sigma^2), \quad \sigma \text{ known for simplicity}$$

- The distribution $p(\mathbf{y}_t|\mathbf{x}_t)$ can be arbitrary (but not depending on θ).
- $\theta = a$ needs to be estimated. Assume a prior $a \sim N(\mu_a, \sigma_a^2)$.
- Can be shown:

$$p(a|\mathbf{x}_{1:t}) = N(\mu_{a|t}, \sigma^2_{a|t})$$

where

$$\mu_{a|t} = \frac{\sigma_a^2 \sum_{s=2}^t \mathbf{x}_s \mathbf{x}_{s-1} + \sigma^2 \mu_a}{\sigma_a^2 \sum_{s=2}^t \mathbf{x}_{s-1}^2 + \sigma^2}; \quad \sigma^2_{a|t} = \frac{\sigma^2 \sigma_a^2}{\sigma_a^2 \sum_{s=2}^t \mathbf{x}_{s-1}^2 + \sigma^2}.$$

Sufficient statistics

- Example:

$$\boldsymbol{x}_t = a\boldsymbol{x}_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim N(0,\sigma^2), \quad \sigma \text{ known for simplicity}$$

- The distribution $p(\boldsymbol{y}_t|\boldsymbol{x}_t)$ can be arbitrary (but not depending on θ).
- $\theta = a$ needs to be estimated. Assume a prior $a \sim N(\mu_a, \sigma_a^2)$.
- Can be shown:

$$p(a|\boldsymbol{x}_{1:t}) = N(\mu_{a|t}, \sigma^2_{a|t})$$

where

$$\mu_{a|t} = \frac{\sigma_a^2 \sum_{s=2}^t \boldsymbol{x}_s\boldsymbol{x}_{s-1} + \sigma^2\mu_a}{\sigma_a^2 \sum_{s=2}^t \boldsymbol{x}_{s-1}^2 + \sigma^2}; \quad \sigma^2_{a|t} = \frac{\sigma^2\sigma_a^2}{\sigma_a^2 \sum_{s=2}^t \boldsymbol{x}_{s-1}^2 + \sigma^2}.$$

- Main point: Given $\boldsymbol{x}_{1:t}$, the distribution of a (and simulation) is simple.
- $p(a|\boldsymbol{x}_{1:t})$ only depend on $S_{t,1} = \sum_{s=2}^t \boldsymbol{x}_s\boldsymbol{x}_{s-1}$ and $S_{t,2} = \sum_{s=2}^t \boldsymbol{x}_{s-1}^2$
- Both terms can be recursively updated through

$$S_{t,1} = S_{t-1,1} + \boldsymbol{x}_t\boldsymbol{x}_{t-1}, \quad S_{t,2} = S_{t-1,2} + \boldsymbol{x}_{t-1}^2.$$

SMC and sufficient statistics

- Assume $p(\boldsymbol{y}_t|\boldsymbol{x}_t)$ do not depend on θ.
- Assume $p(\theta|\boldsymbol{x}_{1:t}) = p(\theta|S_t)$, S_t sufficient statistic.
- Assume $S_t = h(S_{t-1}, \boldsymbol{x}_{t-1}, \boldsymbol{x}_t)$

SMC and sufficient statistics

- Assume $p(\boldsymbol{y}_t|\boldsymbol{x}_t)$ do not depend on θ.
- Assume $p(\theta|\boldsymbol{x}_{1:t}) = p(\theta|S_t)$, S_t sufficient statistic.
- Assume $S_t = h(S_{t-1}, \boldsymbol{x}_{t-1}, \boldsymbol{x}_t)$
- Fearnhead (2002) and Storvik (2002): Focus on $p(\boldsymbol{x}_t, S_t|\boldsymbol{y}_{1:t})$, not $p(\boldsymbol{x}_t, \theta|\boldsymbol{y}_{1:t})$.

SMC and sufficient statistics

- Assume $p(\boldsymbol{y}_t|\boldsymbol{x}_t)$ do not depend on θ.
- Assume $p(\theta|\boldsymbol{x}_{1:t}) = p(\theta|S_t)$, S_t sufficient statistic.
- Assume $S_t = h(S_{t-1}, \boldsymbol{x}_{t-1}, \boldsymbol{x}_t)$
- Fearnhead (2002) and Storvik (2002): Focus on $p(\boldsymbol{x}_t, S_t|\boldsymbol{y}_{1:t})$, not $p(\boldsymbol{x}_t, \theta|\boldsymbol{y}_{1:t})$.
- Assume a properly weighted sample $\{(\boldsymbol{x}_{t-1}^i, S_{t-1}^i, w_{t-1}^i), i = 1, \ldots, N\}$ with respect to $p(\boldsymbol{x}_{t-1}, S_{t-1}|\boldsymbol{y}_{1:t-1})$
- Similar recursions as before:

$$
\begin{aligned}
p(\boldsymbol{x}_t, S_t|\boldsymbol{y}_{1:t-1}) &= \int_{\boldsymbol{x}_{t-1}} p(\boldsymbol{x}_t, S_t|\boldsymbol{x}_{t-1}, S_{t-1})p(\boldsymbol{x}_{t-1}, S_{t-1}|\boldsymbol{y}_{1:t-1})d\boldsymbol{x}_{t-1}dS_{t-1} \\
&\approx \sum_{i=1}^{N} w_{t-1}^i p(\boldsymbol{x}_t, S_t|\boldsymbol{x}_{t-1}^i, S_{t-1}^i) \\
p(\boldsymbol{x}_t, S_t|\boldsymbol{y}_{1:t}) &\approx c \cdot \sum_{i=1}^{N} w_{t-1}^i p(\boldsymbol{x}_t, S_t|\boldsymbol{x}_{t-1}^i, S_{t-1}^i)p(\boldsymbol{y}_t|\boldsymbol{x}_t).
\end{aligned}
$$

SMC and sufficient statistics

- Assume $p(\boldsymbol{y}_t|\boldsymbol{x}_t)$ do not depend on θ.
- Assume $p(\theta|\boldsymbol{x}_{1:t}) = p(\theta|S_t)$, S_t sufficient statistic.
- Assume $S_t = h(S_{t-1}, \boldsymbol{x}_{t-1}, \boldsymbol{x}_t)$
- Fearnhead (2002) and Storvik (2002): Focus on $p(\boldsymbol{x}_t, S_t|\boldsymbol{y}_{1:t})$, not $p(\boldsymbol{x}_t, \theta|\boldsymbol{y}_{1:t})$.
- Assume a properly weighted sample $\{(\boldsymbol{x}_{t-1}^i, S_{t-1}^i, w_{t-1}^i), i = 1, \ldots, N\}$ with respect to $p(\boldsymbol{x}_{t-1}, S_{t-1}|\boldsymbol{y}_{1:t-1})$
- Similar recursions as before:

$$
\begin{aligned}
p(\boldsymbol{x}_t, S_t|\boldsymbol{y}_{1:t-1}) &= \int_{\boldsymbol{x}_{t-1}} p(\boldsymbol{x}_t, S_t|\boldsymbol{x}_{t-1}, S_{t-1})p(\boldsymbol{x}_{t-1}, S_{t-1}|\boldsymbol{y}_{1:t-1})d\boldsymbol{x}_{t-1}dS_{t-1}\\
&\approx \sum_{i=1}^N w_{t-1}^i p(\boldsymbol{x}_t, S_t|\boldsymbol{x}_{t-1}^i, S_{t-1}^i)
\end{aligned}
$$

$$
p(\boldsymbol{x}_t, S_t|\boldsymbol{y}_{1:t}) \approx c \cdot \sum_{i=1}^N w_{t-1}^i p(\boldsymbol{x}_t, S_t|\boldsymbol{x}_{t-1}^i, S_{t-1}^i)p(\boldsymbol{y}_t|\boldsymbol{x}_t).
$$

- Simulation from $p(\boldsymbol{x}_t, S_t|\boldsymbol{x}_{t-1}^i, S_{t-1}^i)$ (possible proposal function)
  1. Simulate $\theta^i \sim p(\theta|\boldsymbol{x}_{t-1}^i, S_{t-1}^i) = p(\theta|S_{t-1}^i)$.
  2. Simulate $\boldsymbol{x}_t^i \sim p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i, \theta^i)$.
  3. Put $S_t^i = h(S_{t-1}^i, \boldsymbol{x}_{t-1}^i, \boldsymbol{x}_t^i)$.

Algorithm (Storvik filter)

Algorithm 6 SMC with parameter updating

Simulate $\theta^i \sim p(\theta)$ for $i = 1, ..., N$. ▷ Initialization
Simulate $\boldsymbol{x}_1^i \sim p(\boldsymbol{x}_1|\theta^i)$ for $i = 1, ..., N$.
Put weights $w_1^i = p(\boldsymbol{y}_1|\boldsymbol{x}_1^i)$.
Put $S_1^i = 0$ for $i = 1, ..., N$.
for $t = 2, 3, ...$ **do** ▷ Sequential Monte Carlo
  Simulate $\theta^i \sim p(\theta|S_{t-1}^i)$ for $i = 1, ..., N$.
  Simulate $\boldsymbol{x}_t^i \sim p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i, \theta^i)$ for $i = 1, ..., N$.
  Put weights $w_t^i = w_{t-1}^i p(\boldsymbol{y}_t|\boldsymbol{x}_t^i)$.
  Put $S_t^i = h(S_{t-1}^i, \boldsymbol{x}_{t-1}^i, \boldsymbol{x}_t^i)$.
  if $\hat{N}_{eff}$ is small **then** ▷ Resampling
    Resample $(\boldsymbol{x}_t^i, S_t^i)$ with probabilities proportional to w_t^i.
    Put $w_t^i = 1/N$.
  end if
end for

`SMC_lin_bin_parest_suff.R`

Offline methods

C. Andrieu, A. Doucet, and R. Holenstein. Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.

N. Chopin, O. Papaspiliopoulos, et al. *An introduction to sequential Monte Carlo*. Springer, 2020.

C. Dai, J. Heng, P. E. Jacob, and N. Whiteley. An invitation to sequential monte carlo samplers. *Journal of the American Statistical Association*, (just-accepted):1–38, 2022.

A. Doucet, N. De Freitas, N. J. Gordon, et al. *Sequential Monte Carlo methods in practice*, volume 1. Springer, 2001.

P. Fearnhead. Markov chain Monte Carlo, sufficient statistics, and particle filters. *Journal of Computational and Graphical Statistics*, 11(4):848–862, 2002.

J. Liu and M. West. Combined parameter and state estimation in simulation-based filtering. In *Sequential Monte Carlo methods in practice*, pages 197–223. Springer, 2001.

C. A. Naesseth, F. Lindsten, T. B. Schön, et al. Elements of sequential monte carlo. *Foundations and Trends® in Machine Learning*, 12(3):307–392, 2019.

G. Poyiadjis, 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*, 98(1):65–80, 2011. doi: 10.1093/biomet/asq062. URL
+http://dx.doi.org/10.1093/biomet/asq062.

G. Storvik. Particle filters for state-space models with the presence of unknown static parameters. *IEEE Transactions on signal Processing*, 50 (2):281–289, 2002.

Resampling

- Degeneracy of weights a serious problem.
- Solution: Resampling (SIR idea)

Resampling

- Degeneracy of weights a serious problem.
- Solution: Resampling (SIR idea)
- How:
  - Resample from $\{\boldsymbol{x}_t^{(1)}, \dots, \boldsymbol{x}_t^{(N)}\}$ with normalized probabilities $w(\boldsymbol{x}_t^{(1)}), \dots, w(\boldsymbol{x}_t^{(N)})$
  - Put all weights equal to 1
  - Either at each time step or when $\widehat{N}_{eff}$ is small

Resampling

- Degeneracy of weights a serious problem.
- Solution: Resampling (SIR idea)
- How:
  - Resample from $\{\boldsymbol{x}_t^{(1)}, \dots, \boldsymbol{x}_t^{(N)}\}$ with normalized probabilities $w(\boldsymbol{x}_t^{(1)}), \dots, w(\boldsymbol{x}_t^{(N)})$
  - Put all weights equal to 1
  - Either at each time step or when $\widehat{N}_{eff}$ is small
- Resampling will introduce extra random noise at the current time-point
- Can reduce noise at later time points

Resampling

- Degeneracy of weights a serious problem.
- Solution: Resampling (SIR idea)
- How:
  - Resample from $\{\boldsymbol{x}_t^{(1)}, ..., \boldsymbol{x}_t^{(N)}\}$ with normalized probabilities $w(\boldsymbol{x}_t^{(1)}), ..., w(\boldsymbol{x}_t^{(N)})$
  - Put all weights equal to 1
  - Either at each time step or when $\widehat{N}_{eff}$ is small
- Resampling will introduce extra random noise at the current time-point
- Can reduce noise at later time points
- Gives a good approximation to $\pi_t(\boldsymbol{x}_t)$
- Does not give a good approximation to $\pi_t(\boldsymbol{x}_{1:t})$ or $\pi(\boldsymbol{x}_1)$!
- `SMC_cosnorm.R`

Hidden Markov models - state space models

- Assume

$$
\begin{aligned}
\boldsymbol{x}_1 &\sim p(\boldsymbol{x}_1) \\
\boldsymbol{x}_t &\sim p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \\
\boldsymbol{x}_t &\sim p(\boldsymbol{y}_t|\boldsymbol{x}_t)
\end{aligned}
$$

- $\{\boldsymbol{y}_t\}$ observed, $\{\boldsymbol{x}_t\}$ hidden
- Aim: $p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t})$ or $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$
- Recursive relationship:

$$
\begin{aligned}
p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t}) &= \frac{p(\boldsymbol{x}_{1:t}, \boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})}{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})} \\
&= \frac{p(\boldsymbol{x}_{1:t-1}|\boldsymbol{y}_{1:t-1})p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{y}_t|\boldsymbol{x}_t)}{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})} \\
&\propto p(\boldsymbol{x}_{1:t-1}|\boldsymbol{y}_{1:t-1})p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{y}_t|\boldsymbol{x}_t)
\end{aligned}
$$

SMC and hidden Markov models

- Assume $g_t(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$

$$
\begin{aligned}
w_t =& \frac{p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t})}{g(\boldsymbol{x}_{1:t})} \\
\propto& \frac{p(\boldsymbol{x}_{1:t-1}|\boldsymbol{y}_{1:t-1})p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{y}_t|\boldsymbol{x}_t)}{p_{\boldsymbol{x}_1}(\boldsymbol{x}_1)\prod_{s=2}^{t} p(\boldsymbol{x}_s|\boldsymbol{x}_{s-1})} \\
=& \frac{p(\boldsymbol{x}_{1:t-1}|\boldsymbol{y}_{1:t-1})}{g(\boldsymbol{x}_{1:t-1})} \frac{p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{y}_t|\boldsymbol{x}_t)}{p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})} \\
=& w_{t-1}p(\boldsymbol{y}_t|\boldsymbol{x}_t)
\end{aligned}
$$

SMC and hidden Markov models

- Assume $g_t(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$

$$
\begin{aligned}
w_t &= \frac{p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t})}{g(\boldsymbol{x}_{1:t})} \\
&\propto \frac{p(\boldsymbol{x}_{1:t-1}|\boldsymbol{y}_{1:t-1})p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{y}_t|\boldsymbol{x}_t)}{p_{\boldsymbol{x}_1}(\boldsymbol{x}_1)\prod_{s=2}^{t} p(\boldsymbol{x}_s|\boldsymbol{x}_{s-1})} \\
&= \frac{p(\boldsymbol{x}_{1:t-1}|\boldsymbol{y}_{1:t-1})}{g(\boldsymbol{x}_{1:t-1})}\,\frac{p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{y}_t|\boldsymbol{x}_t)}{p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})} \\
&= w_{t-1}p(\boldsymbol{y}_t|\boldsymbol{x}_t)
\end{aligned}
$$

- Algorithm
  1. Sample $\boldsymbol{x}_1^i \sim p_{\boldsymbol{x}_1}(\cdot)$, $i = 1, ..., N$.
  2. Let $w_1^{*i} = p(\boldsymbol{y}_1|\boldsymbol{x}_1^i)$, normalize to $w_i^i = w_1^{*i}/\sum_j w_1^{*j}$. Set $t = 2$
  3. Sample $\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i \sim p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i)$, $i = 1, ..., N$.
  4. Append $\boldsymbol{x}_t^i$ to $\boldsymbol{x}_{1:t-1}^i$, obtaining $\boldsymbol{x}_t^i$
  5. Let $w_t^{*i} = w_{t-1}^i p(\boldsymbol{y}_t|\boldsymbol{x}_t^i)$, normalize to $w_t^i = w_t^{*i}/\sum_j w_t^{*j}$.
  6. If $\hat{N}_{eff}$ is small, perform resampling
  7. Increment t and return to step 3

Terrain navigation

- Assume movement model for airplane

$$\begin{aligned}
\boldsymbol{x}_t =& \boldsymbol{x}_{t-1} + \boldsymbol{d}_t + \boldsymbol{\varepsilon}_t \\
\boldsymbol{d}_t =& \text{Drift of plane measured by internal} \\
& \text{navigation system (assumed known)} \\
\boldsymbol{\varepsilon}_t =& \boldsymbol{R}_t^T \boldsymbol{Z}_t \\
\boldsymbol{R}_t =& \frac{1}{\boldsymbol{x}_{1,t-1}^2 + \boldsymbol{x}_{2,t-1}^2} \begin{pmatrix} -\boldsymbol{x}_{1,t-1} & \boldsymbol{x}_{2,t-1} \\ -\boldsymbol{x}_{2,t-1} & -\boldsymbol{x}_{1,t-1} \end{pmatrix} \\
\boldsymbol{Z}_t \sim& N_2\left(\boldsymbol{0}, q^2 \begin{pmatrix} 1 & 0 \\ 0 & k^2 \end{pmatrix}\right) \qquad q = 400, k = 0.5 \\
Y_t =& m(\boldsymbol{x}_t) + \delta_t \\
m(\boldsymbol{x}_t) =& \text{Elevation at point } \boldsymbol{x}_t
\end{aligned}$$

- `Example_6_7.R`

SMC and particle filters

- SMC with resampling usually called particle filters
- Some mix/confusion about terminology, mainly the same!
- Bootstrap filter: SMC for hidden Markov models with $g(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$

Example optimal resampling

- $N = 5$, $\boldsymbol{w} = (0.3, 0.4, 0.05, 0.15, 0.2)$

Example optimal resampling

- $N = 5$, $\boldsymbol{w} = (0.3, 0.4, 0.05, 0.15, 0.2)$
- $N * \boldsymbol{w} = (1.5, 2.0, 0.25, 0.75, 1.0)$

Example optimal resampling

- $N = 5$, $\boldsymbol{w} = (0.3, 0.4, 0.05, 0.15, 0.2)$
- $N * \boldsymbol{w} = (1.5, 2.0, 0.25, 0.75, 1.0)$
- $\tilde{\boldsymbol{N}} = (1, 2, 0, 0, 1)$

Example optimal resampling

- $N = 5$, $\boldsymbol{w} = (0.3, 0.4, 0.05, 0.15, 0.2)$
- $N * \boldsymbol{w} = (1.5, 2.0, 0.25, 0.75, 1.0)$
- $\tilde{\boldsymbol{N}} = (1, 2, 0, 0, 1)$
- $K = 5 - 4 = 1$

Example optimal resampling

- $N = 5$, $\boldsymbol{w} = (0.3, 0.4, 0.05, 0.15, 0.2)$
- $N * \boldsymbol{w} = (1.5, 2.0, 0.25, 0.75, 1.0)$
- $\tilde{\boldsymbol{N}} = (1, 2, 0, 0, 1)$
- $K = 5 - 4 = 1$
- $\tilde{\boldsymbol{N}}/N = (0.2, 0.4, 0.0, 0.0, 0.2)$

Example optimal resampling

- $N = 5$, $\boldsymbol{w} = (0.3, 0.4, 0.05, 0.15, 0.2)$
- $N * \boldsymbol{w} = (1.5, 2.0, 0.25, 0.75, 1.0)$
- $\tilde{\boldsymbol{N}} = (1, 2, 0, 0, 1)$
- $K = 5 - 4 = 1$
- $\tilde{\boldsymbol{N}}/N = (0.2, 0.4, 0.0, 0.0, 0.2)$
- $\boldsymbol{\delta} = (0.1, 0.0, 0.05, 0.15, 0.0)$

Example optimal resampling

- $N = 5$, $\boldsymbol{w} = (0.3, 0.4, 0.05, 0.15, 0.2)$
- $N * \boldsymbol{w} = (1.5, 2.0, 0.25, 0.75, 1.0)$
- $\tilde{\boldsymbol{N}} = (1, 2, 0, 0, 1)$
- $K = 5 - 4 = 1$
- $\tilde{\boldsymbol{N}}/N = (0.2, 0.4, 0.0, 0.0, 0.2)$
- $\boldsymbol{\delta} = (0.1, 0.0, 0.05, 0.15, 0.0)$
- Sample $\boldsymbol{D}$ from Multinom$(1 : N, 1, (\frac{0.1}{0.3}, \frac{0.0}{0.3}, \frac{0.05}{0.3}, \frac{0.15}{0.3}, \frac{0.0}{0.3}))$
  e.g $\boldsymbol{D} = (1, 0, 0, 0, 0)$

Example optimal resampling

- $N = 5$, $\boldsymbol{w} = (0.3, 0.4, 0.05, 0.15, 0.2)$
- $N * \boldsymbol{w} = (1.5, 2.0, 0.25, 0.75, 1.0)$
- $\tilde{\boldsymbol{N}} = (1, 2, 0, 0, 1)$
- $K = 5 - 4 = 1$
- $\tilde{\boldsymbol{N}}/N = (0.2, 0.4, 0.0, 0.0, 0.2)$
- $\boldsymbol{\delta} = (0.1, 0.0, 0.05, 0.15, 0.0)$
- Sample $\boldsymbol{D}$ from Multinom$(1 : N, 1, (\frac{0.1}{0.3}, \frac{0.0}{0.3}, \frac{0.05}{0.3}, \frac{0.15}{0.3}, \frac{0.0}{0.3}))$ e.g $\boldsymbol{D} = (1, 0, 0, 0, 0)$
- Put $\boldsymbol{N} = \tilde{\boldsymbol{N}} + \boldsymbol{D} = (2, 2, 0, 0, 1)$