Advanced Simulation - Lecture 14

March 3rd, 2020
Outline

• Sequential Importance Sampling.

• Resampling step.

• Sequential Monte Carlo / Particle Filters.
Hidden Markov Models

\[ p_\theta(x_1:T, y_1:T) = \mu_\theta(x_1) \prod_{t=2}^{T} f_\theta(x_t | x_{t-1}) g_\theta(y_t | x_t). \]
Sequential Importance Sampling: algorithm

• **At time** $t = 1$
  - Sample $X_1^i \sim q_1(\cdot)$.
  - Compute the weights
    \[
    w_1^i = \frac{\mu(X_1^i)g(y_1 \mid X_1^i)}{q_1(X_1^i)}.
    \]

• **At time** $t \geq 2$
  - Sample $X_t^i \sim q_{t|t-1}(\cdot \mid X_{t-1}^i)$.
  - Compute the weights
    \[
    w_t^i = w_{t-1}^i \times \omega_t^i = w_{t-1}^i \times \frac{f \left( X_t^i \mid X_{t-1}^i \right) g(y_t \mid X_t^i)}{q_{t|t-1}(X_t^i \mid X_{t-1}^i)}.
    \]
Sequential Importance Sampling: prior proposal

- Default choice of proposal:

\[
q_1(x_1) = \mu(x_1), \\
q_{t|t-1}(x_t \mid x_{t-1}) = f(x_t \mid x_{t-1}).
\]

- Then the incremental weight takes the form

\[
\omega(x_{t-1}, x_t) = g(y_t \mid x_t).
\]

- This proposal blindly propagates \( x_{t-1} \) to \( x_t \) without taking \( y_t \) into account.

- We can implement SIS as soon as we can sample from the hidden process \( (X_t)_{t \geq 1} \) and evaluate \( g(y \mid x) \) pointwise.
Sequential Importance Sampling: optimal proposals

- Proposal \( q_{t|t-1}(x_t|x_{t-1}) \) that minimizes the variance of \((\omega_t^i)_{i=1}^{N}\).

- Turns out to be

\[
q_{t|t-1}^{\text{opt}}(x_t|x_{t-1}) = p(x_t|x_{t-1}, y_t) = \frac{f(x_t|x_{t-1})g(y_t|x_t)}{p(y_t|x_{t-1})}.
\]

- This uses the observation \( y_t \) to guide the propagation of \( x_t \).

- Associated incremental weight:

\[
\omega_{t}^{\text{opt}}(x_{t-1}, x_t) = p(y_t|x_{t-1}),
\]

does not depend on \( x_t \).
Sequential Importance Sampling: example

Figure: Estimation of filtering means $\mathbb{E}(x_t \mid y_{1:t})$. 
Sequential Importance Sampling: example

Figure: Estimation of filtering variances $\nabla (x_t | y_{1:t})$. 
Sequential Importance Sampling: example

Figure: Estimation of marginal log likelihoods $\log p(y_{1:t})$. 
Sequential Importance Sampling: example

Figure: Effective sample size over time.
Sequential Importance Sampling: example

Figure: Spread of 100 paths drawn from the prior proposal, and KF means in blue. Darker lines indicate higher weights.
Resampling

- Idea: at time $t$, select particles with high weights, and remove particles with low weights.

- Spend the fixed computational budget “$N$” on the most promising paths.

- Obtain an equally weighted sample $(N^{-1}, \bar{X}^i)$ from a weighted sample $(w^i, X^i)$.

- Resampling on empirical probability measures: input

$$\pi^N(x) = \left( \sum w^j \right)^{-1} \sum w^i \delta_{X^i}(x)$$

and output

$$\bar{\pi}^N(x) = N^{-1} \sum \delta_{\bar{X}^i}(x).$$
Multinomial resampling

- How to draw from an empirical probability distribution?

\[
\pi^N(x) = \frac{1}{\sum_{j=1}^N w^j} \sum_{i=1}^N w^i \delta_{X^i}(x)
\]

- Remember how to draw from a mixture model?

\[
\sum_{i=1}^K \omega^i \ p^i(x)
\]

- Draw \( k \) with probabilities \( \omega^1, \ldots, \omega^N \), then draw from \( p^k \).
Multinomial resampling

- Draw an “ancestry vector”
  \[ A^{1:N} = (A^1, \ldots, A^N) \in \{1, \ldots, N\}^N \] independently from a categorical distribution

  \[ A^{1:N} \overset{i.i.d.}\sim \text{Cat} \left( w^1, \ldots, w^N \right) , \]

  in other words

  \[
  \forall i \in \{1, \ldots, N\} \quad \forall k \in \{1, \ldots, N\} \quad P[A^i = k] = \frac{w^k}{\sum_{j=1}^{N} w^j} .
  \]

- Define \( \bar{X}^i \) to be \( X^{A^i} \) for all \( i \in \{1, \ldots, N\} \). \( X^{A^i} \) is said to be the “parent” or “ancestor” of \( \bar{X}^i \).

- Return \( \bar{X} = (\bar{X}^1, \ldots, \bar{X}^N) \).
Multinomial resampling

- Draw an “offspring vector”
  \[ O^{1:N} = (O^{1}, \ldots, O^{N}) \in \{0, \ldots, N\}^{N} \]
  from a multinomial distribution

  \[ O^{1:N}_{t} \sim \text{Multinomial} \left( N; w^{1}, \ldots, w^{N} \right) \]

  so that

  \[ \forall i \in \{1, \ldots, N\} \quad \mathbb{E}[O^{i}] = N \frac{w^{i}}{\sum_{j=1}^{N} w^{j}} \quad \text{and} \quad \sum_{i=1}^{N} O^{i} = N. \]

- Each particle \( X^{i} \) is replicated \( O^{i} \) times (possibly zero times) to create the sample \( \bar{X} \):
  - \( \bar{X} \leftarrow \{\} \)
  - For \( i = 1, \ldots, N \), for \( k = 0, \ldots, O^{i}_{t} \), \( \bar{X} \leftarrow \{\bar{X}, X^{i}\} \)
- Return \( \bar{X} = (\bar{X}^{1}, \ldots, \bar{X}^{N}) \).
Multinomial resampling

- Other strategies are possible to perform resampling.
- Some properties are desirable:

\[
\mathbb{E}[O^i] = N \frac{w^i}{\sum_{j=1}^{N} w^j},
\]

or \( \mathbb{P}[A^i = k] = \frac{w^k}{\sum_{j=1}^{N} w^j} \).

- This is sometimes called “unbiasedness”, because then

\[
\frac{1}{N} \sum_{k=1}^{N} \varphi(\bar{X}^k) = \frac{1}{N} \sum_{k=1}^{N} O^k \varphi(X^k)
\]

has expectation

\[
\sum_{k=1}^{N} \frac{w^k}{\sum_{j=1}^{N} w^j} \varphi(X^k).
\]
Sequential Monte Carlo: algorithm

- At time $t = 1$
  - Sample $X^i_1 \sim q_1(\cdot)$.
  - Compute the weights
    \[
    w^i_1 = \frac{\mu(X^i_1)g(y_1 | X^i_1)}{q_1(X^i_1)}.
    \]

- At time $t \geq 2$
  - Resample $\left( w^i_{t-1}, X^i_{1:t-1} \right) \rightarrow \left( N^{-1}, \bar{X}^i_{1:t-1} \right)$.
  - Sample $X^i_t \sim q_{t|t-1}(\cdot | \bar{X}^i_{t-1})$, $X^i_{1:t} := \left( \bar{X}^i_{1:t-1}, X^i_t \right)$
  - Compute the weights
    \[
    w^i_t = \omega^i_t = \frac{f \left( X^i_t \mid X^i_{t-1} \right) g(y_t \mid X^i_t)}{q_{t|t-1}(X^i_t \mid X^i_{t-1})}.
    \]
Sequential Monte Carlo: example

Figure: Estimation of filtering means $\mathbb{E}(x_t | y_{1:t})$. 
Sequential Monte Carlo: example

Figure: Estimation of filtering variances $\mathbb{V}(x_t | y_{1:t})$. 
Sequential Monte Carlo: example

Figure: Estimation of marginal log likelihoods $\log p(y_{1:t})$. 
Sequential Monte Carlo: example

![Effective sample size over time.](#)

Figure: Effective sample size over time.
Sequential Monte Carlo: example

Figure: Support of the approximation of $p(x_t | y_{1:t})$, over time.
Sequential Importance Sampling: algorithm

- At time \( t = 1 \)
  - Sample \( X^i_1 \sim q_1(\cdot) \).
  - Compute the weights
    \[
    w^i_1 = \frac{\mu(X^i_1)g(y_1 | X^i_1)}{q_1(X^i_1)}.
    \]

- At time \( t \geq 2 \)
  - Sample \( X^i_t \sim q_{t|t-1}(\cdot | X^i_{t-1}) \), \( X^i_{1:t} := (X^i_{1:t-1}, X^i_t) \).
  - Compute the weights
    \[
    w^i_t = w^i_{t-1} \times \omega^i_t
    = w^i_{t-1} \times \frac{f(X^i_t | X^i_{t-1}) g(y_t | X^i_t)}{q_{t|t-1}(X^i_t | X^i_{t-1})}.
    \]
Sequential Monte Carlo: algorithm

- **At time** $t = 1$
  - Sample $X_1^i \sim q_1(\cdot)$.
  - Compute the weights
    \[
    w_1^i = \frac{\mu(X_1^i)g(y_1 \mid X_1^i)}{q_1(X_1^i)}.
    \]

- **At time** $t \geq 2$
  - Resample $(w_{t-1}^i, X_{1:t-1}^i) \rightarrow (N^{-1}, \overline{X}_{1:t-1}^i)$.
  - Sample $X_t^i \sim q_{t|t-1}(\cdot \mid \overline{X}_{t-1}^i)$, $X_{1:t}^i := (\overline{X}_{1:t-1}^i, X_t^i)$.
  - Compute the weights
    \[
    w_t^i = \omega_t^i = \frac{f(X_t^i \mid X_{t-1}^i)g(y_t \mid X_t^i)}{q_{t|t-1}(X_t^i \mid X_{t-1}^i)}.
    \]
Path degeneracy: example

Figure: Support of the approximation $(\tilde{X}_t^i)_{i=1}^N$ of $p(x_t | y_{1:t})$, over time. The blue curve shows the expectation $\mathbb{E}(x_t | y_{1:t})$ at all times $t$. 
Path degeneracy: example

Figure: Trajectories $\bar{X}_{1:t}^i$, at time $t = 10$. 
Path degeneracy: example

Figure: Trajectories $\bar{X}_{i..t}$, at time $t = 20$. 
Path degeneracy: example

Figure: Trajectories $\bar{X}_{1:t}^i$, at time $t = 30$. 
Path degeneracy: example

Figure: Trajectories $\bar{X}_1^i, t$, at time $t = 40$. 
Path degeneracy: example

Figure: Trajectories $\bar{X}_{i,t}$, at time $t = 50$. 
Path degeneracy: example

Figure: Trajectories $\bar{X}_i^t$, at time $t = 60$. 
Path degeneracy: example

Figure: Trajectories $\bar{X}^i_{1:t}$, at time $t = 70$. 
Path degeneracy: example

Figure: Trajectories $\bar{X}_i^{\cdot t}$, at time $t = 80$. 
Path degeneracy: example

Figure: Trajectories $\bar{X}^i_{1:t}$, at time $t = 90$. 
Path degeneracy: example

Figure: Trajectories $\bar{X}_{1:t}^i$, at time $t = 100$. 
Path degeneracy: output

- Particle approximation of filtering $p(x_t \mid y_{1:t}, \theta)$:

$$\frac{1}{\sum_{j=1}^{N} w_t^j} \sum_{i=1}^{N} w_t^i \delta x_t^i (dx_t),$$

or, after resampling,

$$\frac{1}{N} \sum_{i=1}^{N} \delta \bar{x}_t^i (dx_t).$$

- Particle approximation of path filtering $p(x_{1:t} \mid y_{1:t}, \theta)$:

$$\frac{1}{\sum_{j=1}^{N} w_t^j} \sum_{i=1}^{N} w_t^i \delta x_{1:t}^i (dx_{1:t}),$$

or, similarly, the one after resampling.
Path degeneracy

- Particle filters approximate well $p(x_t \mid y_{1:t})$ but not $p(x_s \mid y_{1:t})$ for $s < < t$.

- Specific particle methods have been developed for this task: fixed lag smoother, forward filtering backward smoothing, etc.

- The simplest is the fixed lag smoother: $p(x_s \mid y_{1:t})$ is approximated by the particle approximation of $p(x_s \mid y_{1:(s+\Delta)\land t})$ for a small integer $\Delta$.

- Fixed-lag smoothing introduces a bias but reduces the variance.
Complexity

- Propagating and weighting the particles is $\mathcal{O}(N)$.
- Each particle can be propagated and weighted in parallel.
- Multinomial resampling is $\mathcal{O}(N)$ if the uniforms are generated in sorted order.
- Resampling cannot be completely parallel, since it creates correlation between the particles.
- The memory cost is $\mathcal{O}(N)$ if only the latest particles are stored.
- The memory cost is at most $\mathcal{O}(Nt)$ if the paths are stored; efficient implementations reduce this to $\mathcal{O}(t + N \log N)$. 
Likelihood estimation

- At time 1,

\[ p^N(y_1) = \frac{1}{N} \sum_{i=1}^{N} w^i \]

\[ \xrightarrow{\text{a.s.}} N \to \infty \int \frac{\mu(x_1)g(y_1 | x_1)}{q_1(x_1)} q_1(x_1) \, dx_1 = p(y_1). \]

- At time \( t \),

\[ p^N(y_t | y_{1:t-1}) = \frac{1}{N} \sum_{i=1}^{N} w^i \]

\[ \xrightarrow{\text{a.s.}} N \to \infty \int w(x_{t-1}, x_t) q_{t|t-1}(x_t | x_{t-1}) p(x_{t-1} | y_{1:t-1}) \, dx_{t-1:t} = p(y_t | y_{1:t-1}). \]

where \( w(x_{t-1}, x_t) = \frac{f(x_t | x_{t-1})g(y_t | x_t)}{q_{t|t-1}(x_t | x_{t-1})} \).
Likelihood estimation

- This leads to the estimator

\[
p^N(y_{1:t}) = p^N(y_1) \prod_{s=2}^{t} p^N(y_s | y_{1:s-1})
\]

\[
= \prod_{s=1}^{t} \frac{1}{N} \sum_{i=1}^{N} w_s^i \xrightarrow{N \to \infty} p(y_{1:t}).
\]

- Surprisingly (?), this estimator is unbiased:

\[
\mathbb{E} \left[ p^N(y_{1:t}) \right] = p(y_{1:t}),
\]

whereas for any \( t \geq 2 \),

\[
\mathbb{E} \left[ p^N(y_t | y_{1:t-1}) \right] \neq p(y_t | y_{1:t-1}).
\]

- Typical particle estimates have a bias of order \( O(1/N) \); the likelihood estimator \( p^N(y_{1:t}) \) is an exception.
Likelihood estimation: example

- Model equations:

\[
\begin{align*}
\forall t \geq 1 \quad X_t &= \varphi X_{t-1} + \sigma_V V_t, \\
\forall t \geq 1 \quad Y_t &= X_t + \sigma_W W_t,
\end{align*}
\]

with \( X_0 \sim \mathcal{N}\left(0, \sigma_V^2\right) \), \( V_t, W_t \) i.i.d. \( \sim \mathcal{N}(0, 1) \), \( \sigma_V = 1 \), \( \sigma_W = 1 \).

- Synthetic data is generated using \( \varphi^* = 0.95 \), and we estimate the likelihood for a range of values of \( \varphi \).
Likelihood estimation: example

Figure: Log-likelihood estimates $\log p^N(y_{1:t} | \varphi)$ as a function of $\varphi$. 12 independent replicates for each value of $\varphi$. 
Selected theoretical results

- Particle filters have been theoretically studied in the past 20 years.

- Consistency as $N \to \infty$ is simple to prove, as each step (propagation, weighting, resampling) is itself consistent.

- Convergence results include Central Limit Theorems and non-asymptotic results.

- They provide guidelines to select the number of particles as a function of $T$, the size of the data, and other algorithmic parameters.
Selected theoretical results

Consider \( I(\varphi_t) = \int \varphi_t(x_{1:t})p(x_{1:t} \mid y_{1:t})dx_{1:t} \).

- \( L_p \)-bound on the path space:

\[
\mathbb{E} \left[ \left| I^N(\varphi_t) - I(\varphi_t) \right|^p \right]^{1/p} \leq \frac{B(t)c(p)\|\varphi_t\|_{\infty}}{\sqrt{N}},
\]

- Central limit theorem on the path space.

\[
\sqrt{N} \left( I^N(\varphi_t) - I(\varphi_t) \right) \xrightarrow{D} \mathcal{N} \left( 0, \sigma^2_t \right),
\]

- As expected, \( B(t) \) and \( \sigma^2_t \) typically grow exponentially fast with \( t \). This is the path degeneracy problem.
Selected theoretical results

Consider instead \( I(\varphi_t) = \int \varphi_t(x_t)p(x_t \mid y_{1:t})dx_t \).

- \( L_p \)-bound:
  \[
  \mathbb{E} \left[ \left| I^N(\varphi_t) - I(\varphi_t) \right|^p \right]^{1/p} \leq \frac{B_1 c(p) \|\varphi_t\|_\infty}{\sqrt{N}} \\
  \sqrt{N} \left( I^N(\varphi_t) - I(\varphi_t) \right) \xrightarrow{\mathcal{D}} \mathcal{N} \left( 0, \sigma_t^2 \right),
  \]

- For the filtering estimates, the error is independent of the time \( t \): \( \sigma_t^2 < \sigma_{\text{max}}^2 \) for all \( t \), and \( B_1 \) independent of \( t \).

- i.e. particle filters are online.
Selected theoretical results

Consider the estimator of the marginal likelihood

\[ p^N(y_{1:t}) = \prod_{s=1}^{t} \frac{1}{N} \sum_{i=1}^{N} w^{i}_s. \]

- Unbiasedness

\[ \mathbb{E} \left[ p^N(y_{1:t}) \right] = p(y_{1:t}). \]

- Non-asymptotic relative variance

\[ \mathbb{E} \left( \left( \frac{p^N(y_{1:t})}{p(y_{1:t})} - 1 \right)^2 \right) \leq \frac{B_3 t}{N}. \]

- Choose \( N = \mathcal{O}(t) \) to control the relative variance.
Next

- Particle Markov chain Monte Carlo (PMCMC) methods, for estimating $\Theta$, using these likelihood estimates.