STAT 576 Bayesian Analysis

Lecture 11: State-space Models and Sequential Monte Carlo II

Chencheng Cai

Washington State University

Sequential Monte Carlo

- Last time, we introduced the state-space models.
- For linear Gaussian state-space models, we can use Kalman filter and smoother to estimate the latent states and parameters.
- The key idea behind the Kalman filter and smoother is to recursively update the filtering and smoothing distributions.

Sequential Monte Carlo

- Last time, we introduced the state-space models.
- For linear Gaussian state-space models, we can use Kalman filter and smoother to estimate the latent states and parameters.
- The key idea behind the Kalman filter and smoother is to recursively update the filtering and smoothing distributions.
- For general state-space models, we usualy do not have closed-form solutions as in the linear Gaussian case.

Sequential Monte Carlo

- Last time, we introduced the state-space models.
- For linear Gaussian state-space models, we can use Kalman filter and smoother to estimate the latent states and parameters.
- The key idea behind the Kalman filter and smoother is to recursively update the filtering and smoothing distributions.
- For general state-space models, we usualy do not have closed-form solutions as in the linear Gaussian case.
- Sequential Monte Carlo (SMC) methods provide a general framework for estimating the filtering and smoothing distributions in general state-space models through Monte Carlo sampling.

The Sequential Structure (MC version)

- In our previous discussion for the Kalman filter and smoother, we have the following recursive structure:

$$X_t \mid \boldsymbol{Y}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{V}_t) \implies X_{t+1} \mid \boldsymbol{Y}_t \sim \mathcal{N}(\boldsymbol{\mu}_{t+1}, \boldsymbol{V}_{t+1}).$$

  It is a consequence of the fact that $(X_{t+1}, Y_{t+1}) \mid X_t$ is multivariate normal.

The Sequential Structure (MC version)

- In our previous discussion for the Kalman filter and smoother, we have the following recursive structure:

$$X_t \mid \boldsymbol{Y}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{V}_t) \implies X_{t+1} \mid \boldsymbol{Y}_t \sim \mathcal{N}(\boldsymbol{\mu}_{t+1}, \boldsymbol{V}_{t+1}).$$

  It is a consequence of the fact that $(X_{t+1}, Y_{t+1}) \mid X_t$ is multivariate normal.
- (**MC version**) Similarly, if we have samples $(\boldsymbol{X}_t^{(i)}, w_t^{(i)})_{i=1}^N$ from the filtering distribution $p(\boldsymbol{X}_t \mid \boldsymbol{Y}_t)$, we can generate samples from the filtering distribution $p(\boldsymbol{X}_{t+1} \mid \boldsymbol{Y}_{t+1})$ by the following steps:

The Sequential Structure (MC version)

- In our previous discussion for the Kalman filter and smoother, we have the following recursive structure:

$$X_t \mid \boldsymbol{Y}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{V}_t) \implies X_{t+1} \mid \boldsymbol{Y}_t \sim \mathcal{N}(\boldsymbol{\mu}_{t+1}, \boldsymbol{V}_{t+1}).$$

  It is a consequence of the fact that $(X_{t+1}, Y_{t+1}) \mid X_t$ is multivariate normal.
- (**MC version**) Similarly, if we have samples $(\boldsymbol{X}_t^{(i)}, w_t^{(i)})_{i=1}^N$ from the filtering distribution $p(\boldsymbol{X}_t \mid \boldsymbol{Y}_t)$, we can generate samples from the filtering distribution $p(\boldsymbol{X}_{t+1} \mid \boldsymbol{Y}_{t+1})$ by the following steps:
  1. Sample $X_{t+1}^{(i)} \sim q_{t+1}(X_{t+1})$ for some proposal distribution q_{t+1}
  2. Let $\boldsymbol{X}_{t+1}^{(i)} = (\boldsymbol{X}_t^{(i)}, X_{t+1}^{(i)})$ and assign weights

$$w_{t+1}^{(i)} = w_t^{(i)} \frac{f_{t+1}(X_{t+1}^{(i)} \mid \boldsymbol{X}_t^{(i)}) g_{t+1}(Y_{t+1} \mid X_{t+1}^{(i)})}{q_{t+1}(X_{t+1}^{(i)})}$$

Sequential Importance Sampling (SIS)

1. **Initialization:**
   1.1 Generate N independent samples $X_0^{(i)}$ from the proposal distribution $q_0(X_0)$.
   1.2 Assign weights $w_0^{(i)} \propto f_0(X_0^{(i)})/q_0(X_0^{(i)})$.
2. **Iteration:** For $t = 1, 2, \ldots, T$,
   2.1 Sample $X_t^{(i)} \sim q_t(X_t)$ for $i = 1, \ldots, N$.
   2.2 Assign weights

$$w_t^{(i)} \propto w_{t-1}^{(i)} \frac{f_t(X_t^{(i)} \mid \boldsymbol{X}_{t-1}^{(i)}) g_t(Y_t \mid X_t^{(i)})}{q_t(X_t^{(i)})}$$

Sequential Importance Sampling (SIS)

1. **Initialization:**
   1.1 Generate N independent samples $X_0^{(i)}$ from the proposal distribution $q_0(X_0)$.
   1.2 Assign weights $w_0^{(i)} \propto f_0(X_0^{(i)})/q_0(X_0^{(i)})$.
2. **Iteration:** For $t = 1, 2, \ldots, T$,
   2.1 Sample $X_t^{(i)} \sim q_t(X_t)$ for $i = 1, \ldots, N$.
   2.2 Assign weights

$$w_t^{(i)} \propto w_{t-1}^{(i)} \frac{f_t(X_t^{(i)} \mid \boldsymbol{X}_{t-1}^{(i)}) g_t(Y_t \mid X_t^{(i)})}{q_t(X_t^{(i)})}$$

Then:

- The weighted samples $(\boldsymbol{X}_t^{(i)}, w_t^{(i)})_{i=1}^N$ are samples from the filtering distribution $p(\boldsymbol{X}_t \mid \boldsymbol{Y}_t)$.
- The weighted samples $(\boldsymbol{X}_T^{(i)}, w_T^{(i)})_{i=1}^N$ are samples from the smoothing distribution $p(\boldsymbol{X}_T \mid \boldsymbol{Y}_{1:T})$.

Justification on the Importance Sampling

- From the principle of impoartance sampling, if $X^{(i)}$ are samples from $q(X)$ and $(X^{(i)}, w^{(i)})$ are (weighted) samples from the target $p(X)$, then

$$w^{(i)} \propto \frac{p(X^{(i)})}{q(X^{(i)})}$$

Justification on the Importance Sampling

- From the principle of impoartance sampling, if $X^{(i)}$ are samples from $q(X)$ and $(X^{(i)}, w^{(i)})$ are (weighted) samples from the target $p(X)$, then

$$w^{(i)} \propto \frac{p(X^{(i)})}{q(X^{(i)})}$$

- For the SIS algorithm, the sampling distributions for $\boldsymbol{X}_t$ is

$$q(\boldsymbol{X}_t) = q_0(X_0) \prod_{s=1}^{t} q_s(X_s)$$

Justification on the Importance Sampling

- From the principle of impoartance sampling, if $X^{(i)}$ are samples from $q(X)$ and $(X^{(i)}, w^{(i)})$ are (weighted) samples from the target $p(X)$, then

$$w^{(i)} \propto \frac{p(X^{(i)})}{q(X^{(i)})}$$

- For the SIS algorithm, the sampling distributions for $\boldsymbol{X}_t$ is

$$q(\boldsymbol{X}_t) = q_0(X_0) \prod_{s=1}^{t} q_s(X_s)$$

- The target filtering distribution is

$$p(\boldsymbol{X}_t \mid \boldsymbol{Y}_t) \propto f_0(X_0) \prod_{s=1}^{t} f_s(X_s \mid \boldsymbol{X}_{s-1}) g_s(Y_s \mid X_s)$$

Justification on the Importance Sampling

- The proper weight for the i-th sample at time t is

$$w_t^{(i)} \propto \frac{p(\boldsymbol{X}_t^{(i)} \mid \boldsymbol{Y}_t)}{q(\boldsymbol{X}_t^{(i)})} \propto \frac{q_0(X_0^{(i)})}{f_0(X_0^{(i)})} \prod_{s=1}^{t} \frac{f_s(X_s^{(i)} \mid \boldsymbol{X}_{s-1}^{(i)}) g_s(Y_s \mid X_s^{(i)})}{q_s(X_s^{(i)})}$$

Justification on the Importance Sampling

- The proper weight for the i-th sample at time t is

$$w_t^{(i)} \propto \frac{p(\boldsymbol{X}_t^{(i)} \mid \boldsymbol{Y}_t)}{q(\boldsymbol{X}_t^{(i)})} \propto \frac{q_0(X_0^{(i)})}{f_0(X_0^{(i)})} \prod_{s=1}^{t} \frac{f_s(X_s^{(i)} \mid \boldsymbol{X}_{s-1}^{(i)}) g_s(Y_s \mid X_s^{(i)})}{q_s(X_s^{(i)})}$$

- On the one hand, this is the cumulated product of the importance weights for the samples up to time t:

$$w_t^{(i)} \propto w_0^{(i)} \prod_{s=1}^{t} \frac{w_s^{(i)}}{w_{s-1}^{(i)}}$$

Justification on the Importance Sampling

- The proper weight for the i-th sample at time t is

$$w_t^{(i)} \propto \frac{p(\boldsymbol{X}_t^{(i)} \mid \boldsymbol{Y}_t)}{q(\boldsymbol{X}_t^{(i)})} \propto \frac{q_0(X_0^{(i)})}{f_0(X_0^{(i)})} \prod_{s=1}^{t} \frac{f_s(X_s^{(i)} \mid \boldsymbol{X}_{s-1}^{(i)}) g_s(Y_s \mid X_s^{(i)})}{q_s(X_s^{(i)})}$$

- On the one hand, this is the cumulated product of the importance weights for the samples up to time t:

$$w_t^{(i)} \propto w_0^{(i)} \prod_{s=1}^{t} \frac{w_s^{(i)}}{w_{s-1}^{(i)}}$$

- On the other hand, the sequential update for the weights is

$$w_t^{(i)} \propto w_{t-1}^{(i)} \frac{f_t(X_t^{(i)} \mid \boldsymbol{X}_{t-1}^{(i)}) g_t(Y_t \mid X_t^{(i)})}{q_t(X_t^{(i)})}$$

Different Choices for the Proposal Distribution

- Particle Filter / Bootstrap Filter:

$$q_t(X_t) = f_t(X_t \mid \boldsymbol{X}_{t-1})$$

- Independent Filter:

$$q_t(X_t) \propto g_t(Y_t \mid X_t)$$

- Conditional Optimal Filter:

$$q_t(X_t) \propto f_t(X_t \mid \boldsymbol{X}_{t-1}) g_t(Y_t \mid X_t)$$

- Auxiliary Particle Filter:

$$q_t(X_t) \propto p(Y_{t+1} \mid X_t)$$

Likelihood Estimation with SIS

Suppose the state-space model dynamics is parametrized by $\boldsymbol{\theta}$ and we want to estimate the likelihood $p(\boldsymbol{Y}_{1:T} \mid \boldsymbol{\theta})$.

Likelihood Estimation with SIS

Suppose the state-space model dynamics is parametrized by $\boldsymbol{\theta}$ and we want to estimate the likelihood $p(\boldsymbol{Y}_{1:T} \mid \boldsymbol{\theta})$.

- The likelihood can be written as a high-dimensional integral:

$$
\begin{aligned}
p(\boldsymbol{Y}_T \mid \boldsymbol{\theta}) &= \int p(\boldsymbol{Y}_T, \boldsymbol{X}_T \mid \boldsymbol{\theta}) d\boldsymbol{X}_T \\
&= \int f_0(X_0 \mid \boldsymbol{\theta}) \prod_{s=1}^{T} f_s(X_s \mid \boldsymbol{X}_{s-1}; \boldsymbol{\theta}) g_s(Y_s \mid X_s; \boldsymbol{\theta}) d\boldsymbol{X}_T
\end{aligned}
$$

- Directly estimate the likelihood is infeasible due to the high-dimensional integral.

Likelihood Estimation with SIS

With SIS, we observe that

$$
\begin{aligned}
\mathbb{E}_{\mathsf{SIS}}\left[\frac{w_t}{w_{t-1}}\right] &= \mathbb{E}_{\mathsf{SIS}}\left[\frac{f_t(X_t \mid \boldsymbol{X}_{t-1};\boldsymbol{\theta})g_t(Y_t \mid X_t;\boldsymbol{\theta})}{q_t(X_t)}\right] \\
&= \int \frac{f_t(X_t \mid \boldsymbol{X}_{t-1};\boldsymbol{\theta})g_t(Y_t \mid X_t;\boldsymbol{\theta})}{q_t(X_t)} q_t(X_t)p(\boldsymbol{X}_{t-1} \mid \boldsymbol{Y}_{t-1};\boldsymbol{\theta})dX_t d\boldsymbol{X}_{t-1} \\
&= \int f_t(X_t \mid \boldsymbol{X}_{t-1};\boldsymbol{\theta})g_t(Y_t \mid X_t;\boldsymbol{\theta})p(\boldsymbol{X}_{t-1} \mid \boldsymbol{Y}_{t-1};\boldsymbol{\theta})dX_t d\boldsymbol{X}_{t-1} \\
&= \int \left(\int f_t(X_t \mid \boldsymbol{X}_{t-1};\boldsymbol{\theta})p(\boldsymbol{X}_{t-1} \mid \boldsymbol{Y}_{t-1};\boldsymbol{\theta})d\boldsymbol{X}_{t-1}\right) g_t(Y_t \mid X_t;\boldsymbol{\theta})dX_t \\
&= \int p(X_t \mid \boldsymbol{Y}_{t-1};\boldsymbol{\theta})g_t(Y_t \mid X_t;\boldsymbol{\theta})dX_t \\
&= p(Y_t \mid \boldsymbol{Y}_{t-1};\boldsymbol{\theta})
\end{aligned}
$$

Likelihood Estimation with SIS

Notice that

$$p(\boldsymbol{Y}_t; \boldsymbol{\theta}) = \prod_{s=1}^{T} p(Y_t \mid \boldsymbol{Y}_{t-1}; \boldsymbol{\theta})$$

Likelihood Estimation with SIS

Notice that

$$p(\mathbf{Y}_t;\boldsymbol{\theta}) = \prod_{s=1}^{T} p(Y_t \mid \mathbf{Y}_{t-1};\boldsymbol{\theta})$$

1. **Initialization:**
   1.1 Set $L = 1$.
   1.2 Generate N independent samples $X_0^{(i)}$ from the proposal distribution $q_0(X_0)$.
   1.3 Assign weights $w_0^{(i)} \propto f_0(X_0^{(i)})/q_0(X_0^{(i)})$.
2. **Iteration:** For $t = 1, 2, \ldots, T$,
   2.1 Sample $X_t^{(i)} \sim q_t(X_t)$ for $i = 1, \ldots, N$.
   2.2 Assign weights

   $$w_t^{(i)} = w_{t-1}^{(i)} \frac{f_t(X_t^{(i)} \mid \boldsymbol{X}_{t-1}^{(i)}) g_t(Y_t \mid X_t^{(i)})}{q_t(X_t^{(i)})}$$

   2.3 Update the likelihood estimate

   $$L = L \cdot \frac{\sum_{i=1}^{N} w_t^{(i)}}{\sum_{i=1}^{N} w_{t-1}^{(i)}}$$

Example

Consider a simple state-space model with the following dynamics:

$$\begin{aligned} X_t \mid X_{t-1} &\sim \mathcal{N}(\phi X_{t-1}, 1) \\ Y_t \mid X_t &\sim \mathcal{N}(X_t, 1) \end{aligned}$$

where ϕ is the parameter to be estimated.

Example

Simulate data from the model with $\phi = 0.6$.

```
T = 20
Y = rep(0, T)
X = 0
for(t in 1:T){
    X = 0.6 * X + rnorm(1)
    Y[t] = X + rnorm(1)
}
```

Example

Compute the likelihood with SIS:

```
llh <- function(phi){
    n = 1000
    x = rep(0, n)
    logw = rep(0, n)
    loglik = 0
    for(t in 1:T){
        z = rnorm(n)/sqrt(2)
        xx = (phi*x + Y[t])/2 + z
        dlogw = -0.5*(xx - phi*x)**2
        dlogw = dlogw - 0.5*(Y[t]-xx)**2
        dlogw = dlogw + z**2
        x = xx
        loglik = loglik + log(sum(exp(logw+dlogw)))
        loglik = loglik - log(sum(exp(logw)))
        logw = logw + dlogw
        logw  = logw - mean(logw)
    }
    return(loglik)
}
```

Example

Compute the MLE:

```
phi.hat = optimize(llh, c(-1, 1), maximum = T)$maximum
```

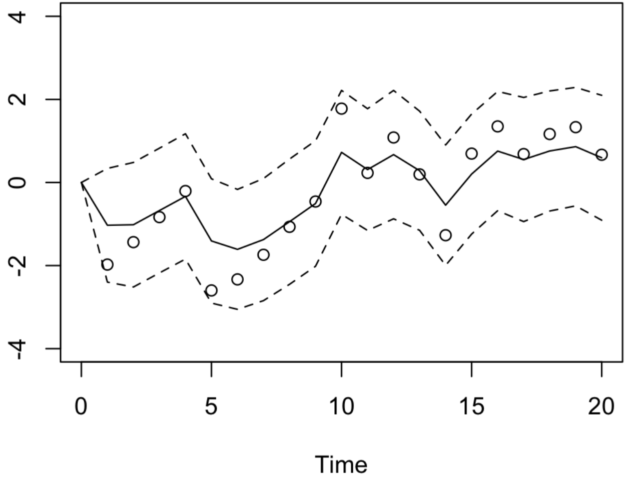
The outcome is $\hat{\phi} = 0.61$. (The result can be noisy due to the randomness in the SIS algorithm and lack of resampling.)

Example

Draw samples from the posterios:

```
smc <- function(phi){
    n = 1000
    x = array(0, c(n, T+1))
    logw = rep(0, n)
    for(t in 1:T){
        z = rnorm(n)/sqrt(2)
        x[,t+1] = (phi*x[,t] + Y[t])/2 + z
        dlogw = -0.5*(x[,t+1] - phi*x[,t])**2
        dlogw = dlogw - 0.5*(Y[t]-x[,t+1])**2
        dlogw = dlogw + z**2
        logw = logw + dlogw
        logw  = logw - mean(logw)
    }
    return(x)
}
```

Example



Example