Sequential Monte Carlo Methods

Isabel Natário[1] M. Lucília Carvalho [2]

[1]Faculdade de Ciências e Tecnologia (UNL); CEAUL

[2]Faculdade de Ciências (UL); CEAUL

March 22, 2010

Introduction

- Bayesian setting

Introduction

- Bayesian setting
- Sequentially arriving observations in time - on line inference (posterior distribution estimation at each step)

Introduction

- Bayesian setting
- Sequentially arriving observations in time - on line inference (posterior distribution estimation at each step)
- **State space models** are used to describe the development over time of an unobserved time series (**the state process**), which is associated with another time-series of observations running in parallel (**the observation process**) - being the relation between the two vectors specified by the state space model

Introduction

- Bayesian setting

- Sequentially arriving observations in time - on line inference (posterior distribution estimation at each step)

- **State space models** are used to describe the development over time of an unobserved time series (**the state process**), which is associated with another time-series of observations running in parallel (**the observation process**) - being the relation between the two vectors specified by the state space model

- For a linear Gaussian state-space model we can get the exact analytical expressions of the posterior distributions - **Kalman filter**

Introduction

- Bayesian setting
- Sequentially arriving observations in time - on line inference (posterior distribution estimation at each step)
- **State space models** are used to describe the development over time of an unobserved time series (**the state process**), which is associated with another time-series of observations running in parallel (**the observation process**) - being the relation between the two vectors specified by the state space model
- For a linear Gaussian state-space model we can get the exact analytical expressions of the posterior distributions - **Kalman filter**
- For a partially observed finite state-space Markov chain we can also get the analytical solution - **Hidden Markov Model (HMM) filter**

- Real data often "suffer" from non-Gaussianity, high dimensionality and non-linearity...

$$\Downarrow$$

- Real data often "suffer" from non-Gaussianity, high dimensionality and non-linearity...

$$\Downarrow$$

- **Sequential Monte Carlo (SMC) methods**

- Real data often "suffer" from non-Gaussianity, high dimensionality and non-linearity...

$$\Downarrow$$

- **Sequential Monte Carlo (SMC) methods**

- Simulation based, flexible, "easy" to implement & applicable in general settings

- Real data often “suffer” from non-Gaussianity, high dimensionality and non-linearity...

$$\Downarrow$$

- **Sequential Monte Carlo (SMC) methods**

- Simulation based, flexible, “easy” to implement & applicable in general settings

- Alias: *Particle filters, Bootstrap filters, condensation, Monte Carlo filters, interacting particle approximations and survival of the fittest*

Framework

- Markovian, non-linear, non-Gaussian state-space models

Framework

- Markovian, non-linear, non-Gaussian state-space models
- **State process:** $\mathbf{x}_t, \ t \in \mathbb{N}_0$ - modeled as a Markov process

Framework

- Markovian, non-linear, non-Gaussian state-space models
- **State process:** $\mathbf{x}_t, \ t \in \mathbb{N}_0$ - modeled as a Markov process
- **Observation process:** $\mathbf{y}_t, \ t \in \mathbb{N}$ - assumed to be conditional independent given the state process

Framework

- Markovian, non-linear, non-Gaussian state-space models
- **State process:** $\mathbf{x}_t, \ t \in \mathbb{N}_0$ - modeled as a Markov process
- **Observation process:** $\mathbf{y}_t, \ t \in \mathbb{N}$ - assumed to be conditional independent given the state process
- Model description:

$$p(\boldsymbol{x}_0)$$
$$p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}), \quad t \geq 1$$
$$p(\boldsymbol{y}_t|\boldsymbol{x}_t), \quad t \geq 1$$

Framework

- Markovian, non-linear, non-Gaussian state-space models
- **State process:** $\mathbf{x}_t, \ t \in \mathbb{N}_0$ - modeled as a Markov process
- **Observation process:** $\mathbf{y}_t, \ t \in \mathbb{N}$ - assumed to be conditional independent given the state process
- Model description:

$$p(\boldsymbol{x}_0)$$
$$p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}), \quad t \geq 1$$
$$p(\boldsymbol{y}_t|\boldsymbol{x}_t), \quad t \geq 1$$

Literature example - non-linear time series model

Framework

- Markovian, non-linear, non-Gaussian state-space models
- **State process:** $\mathbf{x}_t,\ t \in \mathbb{N}_0$ - modeled as a Markov process
- **Observation process:** $\mathbf{y}_t,\ t \in \mathbb{N}$ - assumed to be conditional independent given the state process
- Model description:

$$p(\boldsymbol{x}_0)$$
$$p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}), \quad t \geq 1$$
$$p(\boldsymbol{y}_t|\boldsymbol{x}_t), \quad t \geq 1$$

Literature example - non-linear time series model

$$\begin{aligned}
x_t &= \frac{x_{t-1}}{2} + 25\frac{x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2t) + u_t \\
y_t &= \frac{x_t^2}{20} + v_t \\
u_t &\sim \mathcal{N}(0,\sigma_u^2), \quad \sigma_u^2 = 10 \\
v_t &\sim \mathcal{N}(0,\sigma_v^2), \quad \sigma_v^2 = 1 \\
x_0 &\sim \mathcal{N}(0,10)
\end{aligned}$$

Literature example - non-linear time series model

-

$$\begin{aligned}
f(x_t|x_{t-1}) &\equiv \mathcal{N}\left(\frac{x_{t-1}}{2} + 25\frac{x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2t), \sigma_u^2\right) \\
g(y_t|x_t) &\equiv \mathcal{N}\left(\frac{x_t^2}{20}, \sigma_v^2\right) \\
u_t &\sim \mathcal{N}(0, \sigma_u^2), \quad \sigma_u^2 = 10 \\
v_t &\sim \mathcal{N}(0, \sigma_v^2), \quad \sigma_v^2 = 1 \\
x_0 &\sim \mathcal{N}(0, 10)
\end{aligned}$$

Literature example - non-linear time series model

$$
\begin{aligned}
f(x_t|x_{t-1}) &\equiv \mathcal{N}\left(\frac{x_{t-1}}{2} + 25\frac{x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2t), \sigma_u^2\right) \\
g(y_t|x_t) &\equiv \mathcal{N}\left(\frac{x_t^2}{20}, \sigma_v^2\right) \\
u_t &\sim \mathcal{N}(0, \sigma_u^2), \quad \sigma_u^2 = 10 \\
v_t &\sim \mathcal{N}(0, \sigma_v^2), \quad \sigma_v^2 = 1 \\
x_0 &\sim \mathcal{N}(0, 10)
\end{aligned}
$$

- Joint distribution:

$$\pi(x_{0:T}, y_{1:T}) = \pi(x_0) \times \prod_{t=1}^{T} f(x_t|x_{t-1}) g(y_t|x_t)$$

Literature example - non-linear time series model

$$\begin{aligned}
f(x_t|x_{t-1}) &\equiv \mathcal{N}\left(\frac{x_{t-1}}{2} + 25\frac{x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2t), \sigma_u^2\right)\\
g(y_t|x_t) &\equiv \mathcal{N}\left(\frac{x_t^2}{20}, \sigma_v^2\right)\\
u_t &\sim \mathcal{N}(0, \sigma_u^2), \quad \sigma_u^2 = 10\\
v_t &\sim \mathcal{N}(0, \sigma_v^2), \quad \sigma_v^2 = 1\\
x_0 &\sim \mathcal{N}(0, 10)
\end{aligned}$$

- Joint distribution:

$$\pi(x_{0:T}, y_{1:T}) = \pi(x_0) \times \prod_{t=1}^{T} f(x_t|x_{t-1})g(y_t|x_t)$$

We can simulate from this distribution!

- **Objective** - estimate recursively in time:

- **Objective** - estimate recursively in time:
  - The posterior distribution $p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$;

- **Objective** - estimate recursively in time:
  - The posterior distribution $p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$;
  - Associated features, including the marginal *filtering distribution*, $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$

- **Objective** - estimate recursively in time:
  - The posterior distribution $p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$;
  - Associated features, including the marginal *filtering distribution*, $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$
  - Expectations $I(f_t) = \mathrm{E}_{p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})}[f_t(\boldsymbol{x}_{0:t})] \triangleq \int f_t(\boldsymbol{x}_{0:t})p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})d\boldsymbol{x}_{0:t}$

- **Objective** - estimate recursively in time:
  - The posterior distribution $p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$;
  - Associated features, including the marginal *filtering distribution*, $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$
  - Expectations $I(f_t) = \mathrm{E}_{p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})}[f_t(\boldsymbol{x}_{0:t})] \triangleq \int f_t(\boldsymbol{x}_{0:t})p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})d\boldsymbol{x}_{0:t}$

    e.g.

    - Conditional mean, $f_t(\boldsymbol{x}_{0:t}) = \boldsymbol{x}_{0:t}$
    - Conditional covariance, $f_t(\boldsymbol{x}_{0:t}) = \boldsymbol{x}_t\boldsymbol{x}_t' - \mathrm{E}_{p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})}[\boldsymbol{x}_t]\mathrm{E}'_{p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})}[\boldsymbol{x}_t]$

- *Posterior distribution (Bayes' theorem)*:

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{p(\mathbf{y}_{1:t})} = \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{\int p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})d\mathbf{x}_{0:t}}$$

- *Posterior distribution (Bayes' theorem)*:

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{p(\mathbf{y}_{1:t})} = \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{\int p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})d\mathbf{x}_{0:t}}$$

- *Recursive formula for posterior distribution:*

$$p(\mathbf{x}_{0:t+1}|\mathbf{y}_{1:t+1}) = p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})\frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1})p(\mathbf{x}_{t+1}|\mathbf{x}_{t})}{p(\mathbf{y}_{t+1}|\mathbf{y}_{1:t})}$$

- *Posterior distribution (Bayes' theorem)*:

$$p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = \frac{p(\boldsymbol{y}_{1:t}|\boldsymbol{x}_{0:t})p(\boldsymbol{x}_{0:t})}{p(\boldsymbol{y}_{1:t})} = \frac{p(\boldsymbol{y}_{1:t}|\boldsymbol{x}_{0:t})p(\boldsymbol{x}_{0:t})}{\int p(\boldsymbol{y}_{1:t}|\boldsymbol{x}_{0:t})p(\boldsymbol{x}_{0:t})d\boldsymbol{x}_{0:t}}$$

- *Recursive formula for posterior distribution:*

$$p(\boldsymbol{x}_{0:t+1}|\boldsymbol{y}_{1:t+1}) = p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})\frac{p(\boldsymbol{y}_{t+1}|\boldsymbol{x}_{t+1})p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)}{p(\boldsymbol{y}_{t+1}|\boldsymbol{y}_{1:t})}$$

- Recursive states marginal distributions:
  - *Prediction*:

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}) = \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})d\boldsymbol{x}_{t-1}$$

- *Posterior distribution (Bayes' theorem)*:

$$p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = \frac{p(\boldsymbol{y}_{1:t}|\boldsymbol{x}_{0:t})p(\boldsymbol{x}_{0:t})}{p(\boldsymbol{y}_{1:t})} = \frac{p(\boldsymbol{y}_{1:t}|\boldsymbol{x}_{0:t})p(\boldsymbol{x}_{0:t})}{\int p(\boldsymbol{y}_{1:t}|\boldsymbol{x}_{0:t})p(\boldsymbol{x}_{0:t})d\boldsymbol{x}_{0:t}}$$

- *Recursive formula for posterior distribution:*

$$p(\boldsymbol{x}_{0:t+1}|\boldsymbol{y}_{1:t+1}) = p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})\frac{p(\boldsymbol{y}_{t+1}|\boldsymbol{x}_{t+1})p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)}{p(\boldsymbol{y}_{t+1}|\boldsymbol{y}_{1:t})}$$

- Recursive states marginal distributions:
  - *Prediction*:

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}) = \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})d\boldsymbol{x}_{t-1}$$

  - *Updating*:

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})}{\int p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})d\boldsymbol{x}_t}$$

- *Posterior distribution (Bayes' theorem)*:

$$p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = \frac{p(\boldsymbol{y}_{1:t}|\boldsymbol{x}_{0:t})p(\boldsymbol{x}_{0:t})}{p(\boldsymbol{y}_{1:t})} = \frac{p(\boldsymbol{y}_{1:t}|\boldsymbol{x}_{0:t})p(\boldsymbol{x}_{0:t})}{\int p(\boldsymbol{y}_{1:t}|\boldsymbol{x}_{0:t})p(\boldsymbol{x}_{0:t})d\boldsymbol{x}_{0:t}}$$

- *Recursive formula for posterior distribution:*

$$p(\boldsymbol{x}_{0:t+1}|\boldsymbol{y}_{1:t+1}) = p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})\frac{p(\boldsymbol{y}_{t+1}|\boldsymbol{x}_{t+1})p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_{t})}{p(\boldsymbol{y}_{t+1}|\boldsymbol{y}_{1:t})}$$

- Recursive states marginal distributions:
  - *Prediction*:

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}) = \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})d\boldsymbol{x}_{t-1}$$

  - *Updating*:

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})}{\int p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})d\boldsymbol{x}_t}$$

  - Simple but complicated! :(

- *Posterior distribution (Bayes' theorem)*:

$$p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = \frac{p(\boldsymbol{y}_{1:t}|\boldsymbol{x}_{0:t})p(\boldsymbol{x}_{0:t})}{p(\boldsymbol{y}_{1:t})} = \frac{p(\boldsymbol{y}_{1:t}|\boldsymbol{x}_{0:t})p(\boldsymbol{x}_{0:t})}{\int p(\boldsymbol{y}_{1:t}|\boldsymbol{x}_{0:t})p(\boldsymbol{x}_{0:t})d\boldsymbol{x}_{0:t}}$$

- *Recursive formula for posterior distribution:*

$$p(\boldsymbol{x}_{0:t+1}|\boldsymbol{y}_{1:t+1}) = p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})\frac{p(\boldsymbol{y}_{t+1}|\boldsymbol{x}_{t+1})p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_{t})}{p(\boldsymbol{y}_{t+1}|\boldsymbol{y}_{1:t})}$$

- Recursive states marginal distributions:
  - *Prediction*:

$$p(\boldsymbol{x}_{t}|\boldsymbol{y}_{1:t-1}) = \int p(\boldsymbol{x}_{t}|\boldsymbol{x}_{t-1})p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})d\boldsymbol{x}_{t-1}$$

  - *Updating*:

$$p(\boldsymbol{x}_{t}|\boldsymbol{y}_{1:t}) = \frac{p(\boldsymbol{y}_{t}|\boldsymbol{x}_{t})p(\boldsymbol{x}_{t}|\boldsymbol{y}_{1:t-1})}{\int p(\boldsymbol{y}_{t}|\boldsymbol{x}_{t})p(\boldsymbol{x}_{t}|\boldsymbol{y}_{1:t-1})d\boldsymbol{x}_{t}}$$

  - Simple but complicated! :(

    Late 80s computers enabled development of numerical integration methods for *Bayesian filtering*

Monte Carlo (MC) methods

- MC methods are not subjected to linearity or Gaussian constrains and they have appealing convergence properties

Monte Carlo (MC) methods

- MC methods are not subjected to linearity or Gaussian constrains and they have appealing convergence properties

- Enable us to approximate the intractable integrals that appeared before as long as we have a large number of samples drawn from the required posterior distributions

Monte Carlo (MC) methods

- MC methods are not subjected to linearity or Gaussian constrains and they have appealing convergence properties
- Enable us to approximate the intractable integrals that appeared before as long as we have a large number of samples drawn from the required posterior distributions
- However we can seldom sample from there - importance sampling

Monte Carlo (MC) methods

- MC methods are not subjected to linearity or Gaussian constrains and they have appealing convergence properties
- Enable us to approximate the intractable integrals that appeared before as long as we have a large number of samples drawn from the required posterior distributions
- However we can seldom sample from there - importance sampling
- Making importance sampling recursive: sequential importance sampling, SIS

Perfect Monte Carlo sampling

- Assume we can simulate N random samples (particles) i.i.d. according to $p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$: $\left\{\boldsymbol{x}_{0:t}^{(i)};\ i = 1, \ldots, N\right\}$

Perfect Monte Carlo sampling

- Assume we can simulate N random samples (particles) i.i.d. according to $p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$: $\left\{\boldsymbol{x}_{0:t}^{(i)};\ i=1,\ldots,N\right\}$
- Empirical estimate of $p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$:

$$\hat{p}_N(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = \frac{1}{N}\sum_{i=1}^{N}\delta_{\boldsymbol{x}_{0:t}^{(i)}}(d\boldsymbol{x}_{0:t}) \quad \text{(discrete)}$$

$$\Downarrow$$

$$\hat{I}_N(f_t) = \int f_t(\boldsymbol{x}_{0:t})\hat{p}_N(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})d\boldsymbol{x}_{0:t} = \frac{1}{N}\sum_{i=1}^{N} f_t\left(\boldsymbol{x}_{0:t}^{(i)}\right)$$

Perfect Monte Carlo sampling

- Assume we can simulate N random samples (particles) i.i.d. according to $p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$: $\left\{\boldsymbol{x}_{0:t}^{(i)};\ i = 1, \ldots, N\right\}$
- Empirical estimate of $p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$:

$$\hat{p}_N(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\boldsymbol{x}_{0:t}^{(i)}}(d\boldsymbol{x}_{0:t}) \quad \text{(discrete)}$$

$$\Downarrow$$

$$\hat{I}_N(f_t) = \int f_t(\boldsymbol{x}_{0:t})\hat{p}_N(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})d\boldsymbol{x}_{0:t} = \frac{1}{N}\sum_{i=1}^{N} f_t\left(\boldsymbol{x}_{0:t}^{(i)}\right)$$

- As long as the posterior variance of $f_t(\boldsymbol{x}_{0:t})$ is finite, this estimate converges almost surely to $I(f_t)$ and it follows an approximate Gaussian distribution - *we can estimate any quantity $I(f_t)$ and the rate of convergency is independent of the dimension of the integrand*

Perfect Monte Carlo sampling

- Assume we can simulate N random samples (particles) i.i.d. according to $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$: $\left\{\mathbf{x}_{0:t}^{(i)};\ i = 1, \ldots, N\right\}$
- Empirical estimate of $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$:

$$\hat{p}_N(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\mathbf{x}_{0:t}^{(i)}}(d\mathbf{x}_{0:t}) \quad \text{(discrete)}$$

$$\Downarrow$$

$$\hat{I}_N(f_t) = \int f_t(\mathbf{x}_{0:t})\hat{p}_N(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})d\mathbf{x}_{0:t} = \frac{1}{N}\sum_{i=1}^{N} f_t\left(\mathbf{x}_{0:t}^{(i)}\right)$$

- As long as the posterior variance of $f_t(\mathbf{x}_{0:t})$ is finite, this estimate converges almost surely to $I(f_t)$ and it follows an approximate Gaussian distribution - *we can estimate any quantity $I(f_t)$ and the rate of convergency is independent of the dimension of the integrand*
- Often it is impossible to sample efficiently from the posterior. MCMC unsuited for recursive problems

Importance sampling

- *A technique that simulates samples under a proposal distribution and then approximates the target distribution by conveniently weighting these samples*

Importance sampling

- *A technique that simulates samples under a proposal distribution and then approximates the target distribution by conveniently weighting these samples*

- Objective: estimate $I(f_t)$, dependent of posterior $p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$

Importance sampling

- *A technique that simulates samples under a proposal distribution and then approximates the target distribution by conveniently weighting these samples*

- Objective: estimate $I(f_t)$, dependent of posterior $p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$

- Arbitrary *importance sampling distribution* (or *proposal distribution* or *importance function*): $\pi(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$

Importance sampling

- *A technique that simulates samples under a proposal distribution and then approximates the target distribution by conveniently weighting these samples*

- Objective: estimate $I(f_t)$, dependent of posterior $p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$

- Arbitrary *importance sampling distribution* (or *proposal distribution* or *importance function*): $\pi(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$

- As long as support of $\pi(\cdot)$ includes the support of $p(\cdot)$ we have that:

$$I(f_t) = \frac{\int f_t(\boldsymbol{x}_{0:t}) w(\boldsymbol{x}_{0:t}) \pi(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) d\boldsymbol{x}_{0:t}}{\int w(\boldsymbol{x}_{0:t}) \pi(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) d\boldsymbol{x}_{0:t}},$$

Importance sampling

- *A technique that simulates samples under a proposal distribution and then approximates the target distribution by conveniently weighting these samples*

- Objective: estimate $I(f_t)$, dependent of posterior $p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$

- Arbitrary *importance sampling distribution* (or *proposal distribution* or *importance function*): $\pi(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$

- As long as support of $\pi(\cdot)$ includes the support of $p(\cdot)$ we have that:

$$I(f_t) = \frac{\int f_t(\boldsymbol{x}_{0:t}) w(\boldsymbol{x}_{0:t}) \pi(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) d\boldsymbol{x}_{0:t}}{\int w(\boldsymbol{x}_{0:t}) \pi(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) d\boldsymbol{x}_{0:t}},$$

  where the *importance weight* is:

$$w(\boldsymbol{x}_{0:t}) = \frac{p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})}{\pi(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})}$$

- Consequently if we can simulate N particles i.i.d. according to $\pi(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$, an MC estimate of $I(f_t)$ is:

- Consequently if we can simulate N particles i.i.d. according to $\pi(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$, an MC estimate of $I(f_t)$ is:

$$\tilde{I}_N(f_t) = \frac{\frac{1}{N}\sum_{i=1}^N f_t\left(\boldsymbol{x}_{0:t}^{(i)}\right) w\left(\boldsymbol{x}_{0:t}^{(i)}\right)}{\frac{1}{N}\sum_{j=1}^N w(\boldsymbol{x}_{0:t}^{(j)})} = \sum_{i=1}^N f_t\left(\boldsymbol{x}_{0:t}^{(i)}\right) \tilde{w}_t^{(i)},$$

- Consequently if we can simulate N particles i.i.d. according to $\pi(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$, an MC estimate of $I(f_t)$ is:

$$\tilde{I}_N(f_t) = \frac{\frac{1}{N}\sum_{i=1}^{N} f_t\left(\boldsymbol{x}_{0:t}^{(i)}\right) w\left(\boldsymbol{x}_{0:t}^{(i)}\right)}{\frac{1}{N}\sum_{j=1}^{N} w(\boldsymbol{x}_{0:t}^{(j)})} = \sum_{i=1}^{N} f_t\left(\boldsymbol{x}_{0:t}^{(i)}\right) \tilde{w}_t^{(i)},$$

where the *normalized importance weight* is $\tilde{w}_t^{(i)} = \frac{w\left(\boldsymbol{x}_{0:t}^{(i)}\right)}{\sum_{j=1}^{N} w\left(\boldsymbol{x}_{0:t}^{(j)}\right)}$

- Consequently if we can simulate N particles i.i.d. according to $\pi(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$, an MC estimate of $I(f_t)$ is:

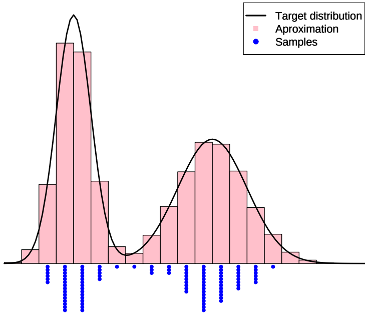
$$\tilde{I}_N(f_t) = \frac{\frac{1}{N}\sum_{i=1}^{N} f_t\left(\boldsymbol{x}_{0:t}^{(i)}\right) w\left(\boldsymbol{x}_{0:t}^{(i)}\right)}{\frac{1}{N}\sum_{j=1}^{N} w(\boldsymbol{x}_{0:t}^{(j)})} = \sum_{i=1}^{N} f_t\left(\boldsymbol{x}_{0:t}^{(i)}\right) \tilde{w}_t^{(i)},$$

  where the *normalized importance weight* is $\tilde{w}_t^{(i)} = \frac{w\left(\boldsymbol{x}_{0:t}^{(i)}\right)}{\sum_{j=1}^{N} w\left(\boldsymbol{x}_{0:t}^{(j)}\right)}$

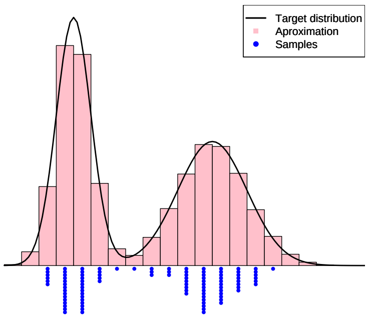
- Under weak conditions, this estimate converges almost surely to $I(f_t)$ and with additional conditions follows an approximate Gaussian distribution - *rate of convergency still independent of the dimension of the integrand*

- This integration method can also be interpreted as a sampling method where $p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$ is approximated by $\tilde{p}_N(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = \sum_{i=1}^{N} \tilde{w}_t^{(i)} \delta_{\boldsymbol{x}_{0:t}^{(i)}}(d\boldsymbol{x}_{0:t})$ and $\tilde{I}_N(f_t)$ is $f_t(\boldsymbol{x}_{0:t})$ integrated with respect to the empirical measure $\tilde{p}_N(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$

- This integration method can also be interpreted as a sampling method where $p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$ is approximated by $\tilde{p}_N(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = \sum_{i=1}^{N} \tilde{w}_t^{(i)} \delta_{\boldsymbol{x}_{0:t}^{(i)}}(d\boldsymbol{x}_{0:t})$ and $\tilde{I}_N(f_t)$ is $f_t(\boldsymbol{x}_{0:t})$ integrated with respect to the empirical measure $\tilde{p}_N(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$



- This integration method can also be interpreted as a sampling method where $p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$ is approximated by $\tilde{p}_N(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = \sum_{i=1}^N \tilde{w}_t^{(i)} \delta_{\boldsymbol{x}_{0:t}^{(i)}}(d\boldsymbol{x}_{0:t})$ and $\tilde{I}_N(f_t)$ is $f_t(\boldsymbol{x}_{0:t})$ integrated with respect to the empirical measure $\tilde{p}_N(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$



- This method is not adequate for recursive estimation, as we need to get all the data to estimate $p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$ - computational complexity increases in time (importance weights).

Sequential importance sampling, SIS

- Modification of importance sampling such that it is possible to get an estimate $\tilde{p}_N(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$ without modifying past trajectories $\left\{\boldsymbol{x}_{0:t-1}^{(i)};\ i = 1, \ldots, N\right\}$

Sequential importance sampling, SIS

- Modification of importance sampling such that it is possible to get an estimate $\tilde{p}_N(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$ without modifying past trajectories $\left\{\boldsymbol{x}_{0:t-1}^{(i)};\ i = 1, \dots, N\right\}$

$$\Downarrow$$

- The importance function $\pi(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$ at t should admit as marginal distribution at $t-1$ the importance function $\pi(\boldsymbol{x}_{0:t-1}|\boldsymbol{y}_{1:t-1})$:

Sequential importance sampling, SIS

- Modification of importance sampling such that it is possible to get an estimate $\tilde{p}_N(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ without modifying past trajectories $\left\{\mathbf{x}_{0:t-1}^{(i)};\ i = 1, \dots, N\right\}$

$$\Downarrow$$

- The importance function $\pi(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ at t should admit as marginal distribution at $t-1$ the importance function $\pi(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})$:

$$\pi(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \pi(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})\pi(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})$$

$$\Downarrow$$

$$\pi(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \pi(\mathbf{x}_0)\prod_{k=1}^{t}\pi(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k})$$

- We can then evaluate recursively in time the important weights:

$$w_t^{(i)} \propto w_{t-1}^{(i)} \frac{p\left(\mathbf{y}_t|\mathbf{x}_t^{(i)}\right) p\left(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}\right)}{\pi\left(\mathbf{x}_t^{(i)}|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{1:t}\right)}$$

Sequential importance sampling, SIS

- Modification of importance sampling such that it is possible to get an estimate $\tilde{p}_N(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$ without modifying past trajectories $\left\{\boldsymbol{x}_{0:t-1}^{(i)};\ i=1,\ldots,N\right\}$

$$\Downarrow$$

- The importance function $\pi(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$ at t should admit as marginal distribution at $t-1$ the importance function $\pi(\boldsymbol{x}_{0:t-1}|\boldsymbol{y}_{1:t-1})$:

$$\pi(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = \pi(\boldsymbol{x}_{0:t-1}|\boldsymbol{y}_{1:t-1})\pi(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1},\boldsymbol{y}_{1:t})$$

$$\Downarrow$$

$$\pi(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = \pi(\boldsymbol{x}_0)\prod_{k=1}^{t}\pi(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1},\boldsymbol{y}_{1:k})$$

Sequential importance sampling, SIS

- Modification of importance sampling such that it is possible to get an estimate $\tilde{p}_N(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$ without modifying past trajectories $\left\{\boldsymbol{x}_{0:t-1}^{(i)};\ i=1,\ldots,N\right\}$

$$\Downarrow$$

- The importance function $\pi(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$ at t should admit as marginal distribution at $t-1$ the importance function $\pi(\boldsymbol{x}_{0:t-1}|\boldsymbol{y}_{1:t-1})$:

$$\pi(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = \pi(\boldsymbol{x}_{0:t-1}|\boldsymbol{y}_{1:t-1})\pi(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1},\boldsymbol{y}_{1:t})$$

$$\Downarrow$$

$$\pi(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = \pi(\boldsymbol{x}_0)\prod_{k=1}^{t}\pi(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1},\boldsymbol{y}_{1:k})$$

- We can then evaluate recursively in time the important weights:

$$w_t^{(i)} \propto w_{t-1}^{(i)} \frac{p\left(\boldsymbol{y}_t|\boldsymbol{x}_t^{(i)}\right) p\left(\boldsymbol{x}_t^{(i)}|\boldsymbol{x}_{t-1}^{(i)}\right)}{\pi\left(\boldsymbol{x}_t^{(i)}|\boldsymbol{x}_{0:t-1}^{(i)},\boldsymbol{y}_{1:t}\right)}$$

- Particular important case - prior distribution as importance distribution:

- Particular important case - prior distribution as importance distribution:

$$\pi(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = p(\boldsymbol{x}_{0:t}) = p(\boldsymbol{x}_0)\prod_{k=1}^{t} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$$

- Particular important case - prior distribution as importance distribution:

$$\pi(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = p(\boldsymbol{x}_{0:t}) = p(\boldsymbol{x}_0)\prod_{k=1}^{t} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$$

- In this case:

$$w_t^{(i)} \propto w_{t-1}^{(i)} p\left(\boldsymbol{y}_t|\boldsymbol{x}_t^{(i)}\right)$$

- Particular important case - prior distribution as importance distribution:

$$\pi(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = p(\boldsymbol{x}_{0:t}) = p(\boldsymbol{x}_0)\prod_{k=1}^{t} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$$

- In this case:

$$w_t^{(i)} \propto w_{t-1}^{(i)} p\left(\boldsymbol{y}_t|\boldsymbol{x}_t^{(i)}\right)$$

- SIS is attractive but is a constrained version of importance sampling, which is usually inefficient in high-dimensional spaces

- Particular important case - prior distribution as importance distribution:

$$\pi(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = p(\boldsymbol{x}_{0:t}) = p(\boldsymbol{x}_0)\prod_{k=1}^{t} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$$

- In this case:

$$w_t^{(i)} \propto w_{t-1}^{(i)} p\left(\boldsymbol{y}_t|\boldsymbol{x}_t^{(i)}\right)$$

- SIS is attractive but is a constrained version of importance sampling, which is usually inefficient in high-dimensional spaces

- The problem is as t increases the distribution of the weights becomes more and more skewed and practically after a few steps only one particle has a non-zero importance weight - need to introduce additional selection step

Sequential Importance Sampling with Resampling

- Idea: eliminate particles with low importance weights and multiplying particles with high weights

Sequential Importance Sampling with Resampling

- Idea: eliminate particles with low importance weights and multiplying particles with high weights

$\Downarrow$

- Replace the weighted empirical dist. $\tilde{p}_N(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = \sum_{j=1}^{N} \tilde{w}_t^{(i)} \delta_{\boldsymbol{x}_{0:t}^{(i)}}(d\boldsymbol{x}_{0:t})$ by the unweighted measure:

$$\hat{p}_N(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = \sum_{j=1}^{N} N_t^{(i)} \delta_{\boldsymbol{x}_{0:t}^{(i)}}(d\boldsymbol{x}_{0:t}),$$

where $N_t^{(i)}$ is the number of offspring associated to particle $\boldsymbol{x}_{0:t}$ - $N = \sum_{i=1}^{N} N_t^{(i)}$; $N_t^{(j)} = 0$ means that particle $\boldsymbol{x}_{0:t}^{(j)}$ dies

Sequential Importance Sampling with Resampling

- Idea: eliminate particles with low importance weights and multiplying particles with high weights

$\Downarrow$

- Replace the weighted empirical dist. $\tilde{p}_N(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = \sum_{j=1}^{N} \tilde{w}_t^{(i)} \delta_{\boldsymbol{x}_{0:t}^{(i)}}(d\boldsymbol{x}_{0:t})$ by the unweighted measure:

$$\hat{p}_N(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = \sum_{j=1}^{N} N_t^{(i)} \delta_{\boldsymbol{x}_{0:t}^{(i)}}(d\boldsymbol{x}_{0:t}),$$

  where $N_t^{(i)}$ is the number of offspring associated to particle $\boldsymbol{x}_{0:t}$ - $N = \sum_{i=1}^{N} N_t^{(i)}$; $N_t^{(j)} = 0$ means that particle $\boldsymbol{x}_{0:t}^{(j)}$ dies

- The $N_t^{(i)}$ are chosen such that $\hat{p}_N(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$ is close to $\tilde{p}_N(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$ in the sense that, for any f_t, its integral with respect to both empirical measures is close

Sequential Importance Sampling with Resampling

- Idea: eliminate particles with low importance weights and multiplying particles with high weights

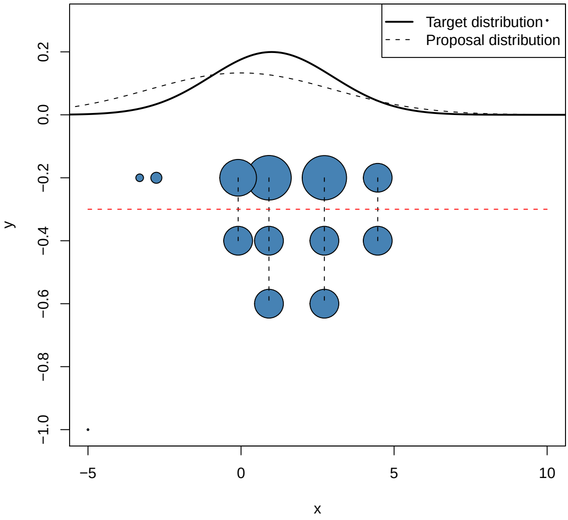
$$\Downarrow$$

- Replace the weighted empirical dist. $\tilde{p}_N(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = \sum_{j=1}^{N} \tilde{w}_t^{(i)} \delta_{\boldsymbol{x}_{0:t}^{(i)}}(d\boldsymbol{x}_{0:t})$ by the unweighted measure:

$$\hat{p}_N(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = \sum_{j=1}^{N} N_t^{(i)} \delta_{\boldsymbol{x}_{0:t}^{(i)}}(d\boldsymbol{x}_{0:t}),$$

  where $N_t^{(i)}$ is the number of offspring associated to particle $\boldsymbol{x}_{0:t}$ - $N = \sum_{i=1}^{N} N_t^{(i)}$; $N_t^{(j)} = 0$ means that particle $\boldsymbol{x}_{0:t}^{(j)}$ dies
- The $N_t^{(i)}$ are chosen such that $\hat{p}_N(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$ is close to $\tilde{p}_N(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$ in the sense that, for any f_t, its integral with respect to both empirical measures is close
- In practice: We obtain the surviving particles by sampling N times from the (discrete) distribution $\tilde{p}_N(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$, which is equivalent to sampling the number of offspring $N_t^{(i)}$ according to a multinomial distribution of parameters $\tilde{w}_t^{(i)}$.

SISR


Target distribution
Proposal distribution
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
y
−5
0
5
10
x

Bootstrap filter algorithm (special particle filter)

1. *Importance sampling steps*
   Generate sample from predictive density of each old sample point $\boldsymbol{x}_{t-1}^{(i)}$:
   $$\tilde{\boldsymbol{x}}_t^{(i)} \sim p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}^{(i)})$$
   Form $\tilde{\boldsymbol{x}}_{1:t}^{(i)} = \left\{ \boldsymbol{x}_{1:t-1}^{(i)}, \tilde{\boldsymbol{x}}_t^{(i)} \right\}$
2. Evaluate importance weights for each new sample point $\tilde{\boldsymbol{x}}_t^{(i)}$:
   $$w_t^{(i)} \sim p(\boldsymbol{y}_t | \tilde{\boldsymbol{x}}_t^{(i)})$$
   Normalize them, $\tilde{w}_t^{(i)} = w_t^{(i)} / \sum_j w_t^{(j)}$
3. *Selection step*
   Resample with replacement by selecting new samples $\boldsymbol{x}_{1:t}^{(i)}$ from the set $\left\{ \tilde{\boldsymbol{x}}_{1:t}^{(i)} \right\}$ with probabilities proportional to $\tilde{w}_t^{(i)}$

Literature example revisited

$$
\begin{aligned}
x_t &= \frac{x_{t-1}}{2} + 25\frac{x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2t) + u_t \\
y_t &= \frac{x_t^2}{20} + v_t \\
u_t &\sim \mathcal{N}(0,\sigma_u^2), \quad \sigma_u^2 = 10 \\
v_t &\sim \mathcal{N}(0,\sigma_v^2), \quad \sigma_v^2 = 1 \\
x_0 &\sim \mathcal{N}(0,10)
\end{aligned}
$$

Literature example revisited

```
####Bootrap filter example:
```

Literature example revisited

```
####Bootrap filter example:
####Observations and true states generation:
v.u<-10
v.v<-1
xT<-rnorm(1,0,sqrt(v.u)) #True state
yT<-NULL
for (t in 1:10){  #Generate state x_t=...  and obs.  y_t=x_t^2/20+vt, vt ~ N(0,vt)
xT<-c(xT,rnorm(1,(xT[t]/2+25*xT[t]/(1+xT[t]^2)+8*cos(1.5*t)),sqrt(v.u)))
yT<-c(yT,rnorm(1,(median(xT[t+1]))^2/20,sqrt(v.v))) }
```

Literature example revisited

```
####Bootrap filter example:
####Observations and true states generation:
v.u<-10
v.v<-1
xT<-rnorm(1,0,sqrt(v.u)) #True state
yT<-NULL
for (t in 1:10){  #Generate state x_t=...  and obs.  y_t=x_t^2/20+vt, vt ~ N(0,vt)
xT<-c(xT,rnorm(1,(xT[t]/2+25*xT[t]/(1+xT[t]^2)+8*cos(1.5*t)),sqrt(v.u)))
yT<-c(yT,rnorm(1,(median(xT[t+1]))^2/20,sqrt(v.v))) }

N<-10000
```

Literature example revisited

```
####Bootrap filter example:
####Observations and true states generation:
v.u<-10
v.v<-1
xT<-rnorm(1,0,sqrt(v.u)) #True state
yT<-NULL
for (t in 1:10){  #Generate state x_t=...  and obs.  y_t=x_t^2/20+vt, vt ~ N(0,vt)
xT<-c(xT,rnorm(1,(xT[t]/2+25*xT[t]/(1+xT[t]^2)+8*cos(1.5*t)),sqrt(v.u)))
yT<-c(yT,rnorm(1,(median(xT[t+1]))^2/20,sqrt(v.v))) }

N<-10000
####Inicialização
x0<-rnorm(N,0,sqrt(v.u))   #p(x_0)~ Normal(0,10)
x.old<-x0
x.todos<-cbind(x.old)
```

Literature example revisited

```
####Bootrap filter example:
####Observations and true states generation:
v.u<-10
v.v<-1
xT<-rnorm(1,0,sqrt(v.u)) #True state
yT<-NULL
for (t in 1:10){  #Generate state x_t=...  and obs.  y_t=x_t^2/20+vt, vt ~ N(0,vt)
xT<-c(xT,rnorm(1,(xT[t]/2+25*xT[t]/(1+xT[t]^2)+8*cos(1.5*t)),sqrt(v.u)))
yT<-c(yT,rnorm(1,(median(xT[t+1]))^2/20,sqrt(v.v))) }

N<-10000
####Inicialização
x0<-rnorm(N,0,sqrt(v.u))   #p(x_0)~ Normal(0,10)
x.old<-x0
x.todos<-cbind(x.old)
for (t in 1:10){
####Importance sampling step
x.new<-rnorm(N,(x.old/2+25*x.old/(1+x.old^2)+8*cos(1.5*t)),sqrt(v.u))
x.todos<-cbind(x.todos,x.new)
```

Literature example revisited

```
####Bootrap filter example:
####Observations and true states generation:
v.u< -10
v.v< -1
xT< -rnorm(1,0,sqrt(v.u)) #True state
yT< -NULL
for (t in 1:10){  #Generate state x_t=... and obs. y_t=x_t^2/20+vt, vt ~ N(0,vt)
xT< -c(xT,rnorm(1,(xT[t]/2+25*xT[t]/(1+xT[t]^2)+8*cos(1.5*t)),sqrt(v.u)))
yT< -c(yT,rnorm(1,(median(xT[t+1]))^2/20,sqrt(v.v))) }

N< -10000
####Inicialização
x0< -rnorm(N,0,sqrt(v.u))   #p(x_0)~ Normal(0,10)
x.old< -x0
x.todos< -cbind(x.old)
for (t in 1:10){
####Importance sampling step
x.new< -rnorm(N,(x.old/2+25*x.old/(1+x.old^2)+8*cos(1.5*t)),sqrt(v.u))
x.todos< -cbind(x.todos,x.new)
#Importance weights
if (t==1){
#Initialize importance weight w_1=p(y1|x1) ~ Normal()
weight< -dnorm(yT[t],x.new^2/20,sqrt(v.v))}
else {
#Calculate importance weight w_t=w_t-1*p(y_t|x_t)
weight.new< -dnorm(yT[t],x.new^2/20,sqrt(v.v)) #No Bootstap o w_t-1 é 1/N, não considerar}
```

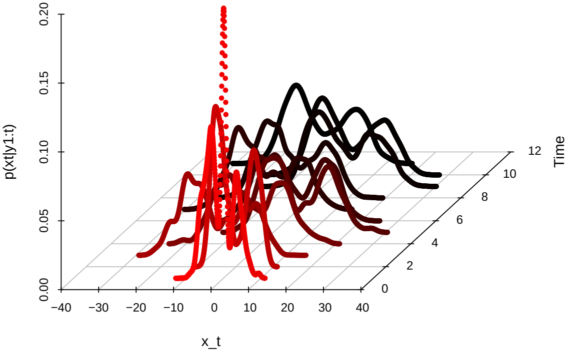
Literature example revisited

```
####Bootrap filter example:
####Observations and true states generation:
v.u<-10
v.v<-1
xT<-rnorm(1,0,sqrt(v.u)) #True state
yT<-NULL
for (t in 1:10){  #Generate state x_t=...  and obs.  y_t=x_t^2/20+vt, vt ~ N(0,vt)
xT<-c(xT,rnorm(1,(xT[t]/2+25*xT[t]/(1+xT[t]^2)+8*cos(1.5*t)),sqrt(v.u)))
yT<-c(yT,rnorm(1,(median(xT[t+1]))^2/20,sqrt(v.v))) }

N<-10000
####Inicialização
x0<-rnorm(N,0,sqrt(v.u))   #p(x_0)~ Normal(0,10)
x.old<-x0
x.todos<-cbind(x.old)
for (t in 1:10){
####Importance sampling step
x.new<-rnorm(N,(x.old/2+25*x.old/(1+x.old^2)+8*cos(1.5*t)),sqrt(v.u))
x.todos<-cbind(x.todos,x.new)
#Importance weights
if (t==1){
#Initialize importance weight w_1=p(y1|x1) ~ Normal()
weight<-dnorm(yT[t],x.new^2/20,sqrt(v.v))}
else {
#Calculate importance weight w_t=w_t-1*p(y_t|x_t)
weight.new<-dnorm(yT[t],x.new^2/20,sqrt(v.v)) #No Bootstap o w_t-1 é 1/N, não considerar}
#Normalize importance weights:
weight<-weight/sum(weight)
```

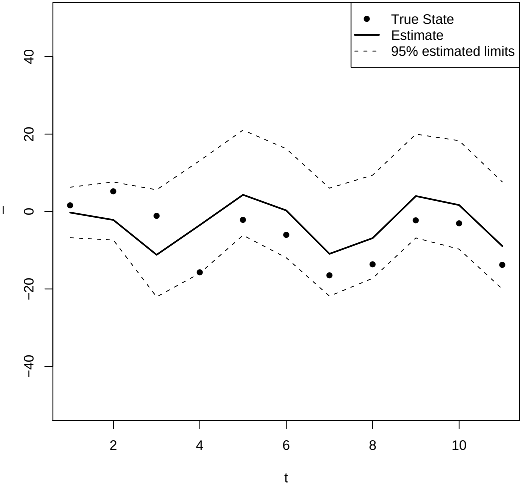
Literature example revisited

```
####Bootrap filter example:
####Observations and true states generation:
v.u< -10
v.v< -1
xT< -rnorm(1,0,sqrt(v.u)) #True state
yT< -NULL
for (t in 1:10){  #Generate state x_t=...  and obs.  y_t=x_t^2/20+vt, vt ~ N(0,vt)
xT< -c(xT,rnorm(1,(xT[t]/2+25*xT[t]/(1+xT[t]^2)+8*cos(1.5*t)),sqrt(v.u)))
yT< -c(yT,rnorm(1,(median(xT[t+1]))^2/20,sqrt(v.v))) }

N< -10000
####Inicialização
x0< -rnorm(N,0,sqrt(v.u))   #p(x_0)~ Normal(0,10)
x.old< -x0
x.todos< -cbind(x.old)
for (t in 1:10){
####Importance sampling step
x.new< -rnorm(N,(x.old/2+25*x.old/(1+x.old^2)+8*cos(1.5*t)),sqrt(v.u))
x.todos< -cbind(x.todos,x.new)
#Importance weights
if (t==1){
#Initialize importance weight w_1=p(y1|x1) ~ Normal()
weight< -dnorm(yT[t],x.new^2/20,sqrt(v.v))}
else {
#Calculate importance weight w_t=w_t-1*p(y_t|x_t)
weight.new< -dnorm(yT[t],x.new^2/20,sqrt(v.v)) #No Bootstap o w_t-1 é 1/N, não considerar}
#Normalize importance weights:
weight< -weight/sum(weight)
####Selection step:
#Sample with replacement N trajectories according to actual importance weights:
index.resamp< -sample(1:N,prob=weight,replace=T)
x.todos< -x.todos[index.resamp,]}
```

Literature example revisited



Literature example revisited



Improvements to SIR

- There are several improvements that have been proposed to SIR methods

Improvements to SIR

- There are several improvements that have been proposed to SIR methods
- For example, to overcome a problem of "particle deplection" (particles with relative large sizes tend to be chosen many times and dominate):
  - Kernel smoothing of parameter vectors at each time step, adding a small perturbation to parameter values, increasing the diversity of parameters values in vicinity of parameter space.

Improvements to SIR

- There are several improvements that have been proposed to SIR methods

- For example, to overcome a problem of "particle deplection" (particles with relative large sizes tend to be chosen many times and dominate):
  - Kernel smoothing of parameter vectors at each time step, adding a small perturbation to parameter values, increasing the diversity of parameters values in vicinity of parameter space.
  - Auxiliary particle filter (an initial "auxiliary" resample is taken from the population at time t, with weights calculated according to the expected likelihood of the states at time $t+1$, given the data as time $t+1$. This resampled set of particles is then projected forward from time t to time $t+1$, and "corrected" using likelihood weights just as with the bootstrap filter, except that the likelihood weights must take account of the auxiliary resampling stage.)

Thank you!

References

- Gordon NJ, Salmond DJ, Smith AFM (1993). *Novel Approach to nonlinea/non-Gaussian Bayesian State Estimation*. IEE Proceedings, 140: 107-113. **(Over 2500 citations!)**

- Doucet A, de Freitas N, Gordon N (Eds.) (2001). *Sequential Monte Carlo in Practice*. Springer.

- Cappé O, Godsill SJ, Moulines E (2007). *An Overview of Exiting Methods and Recent Advances in Sequential Monte Carlo*. Proceedings of the IEEE, 95: 899-922.

- Doucet A, Johansen AM (2008). *A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later*.