A Bayesian Tutorial for Data Assimilation: Christopher K. Wikle & L. Mark Berliner

Damon McDougall

University of Warwick Mathematics Institute

March 16, 2011

Overview

- Introduction

Overview

- Introduction
- Bayesian Inference

Overview

- Introduction
- Bayesian Inference
- Sequential Approaches

Overview

- Introduction
- Bayesian Inference
- Sequential Approaches
- Monte Carlo Sampling and Data Assimilation

Overview

- Introduction
- Bayesian Inference
- Sequential Approaches
- Monte Carlo Sampling and Data Assimilation
- Summary

Introduction

- What is DA?

Introduction

- What is DA?
  - Various definitions

Introduction

- What is DA?
  - Various definitions
  - Their definition: "DA is an approach for fusing data (observations) with prior knowledge to obtain an estimate of the distribution of the true state of the process"

Introduction

- What is DA?
    - Various definitions
    - Their definition: "DA is an approach for fusing data (observations) with prior knowledge to obtain an estimate of the distribution of the true state of the process"
- What do we need?

Introduction

- What is DA?
  - Various definitions
  - Their definition: "DA is an approach for fusing data (observations) with prior knowledge to obtain an estimate of the distribution of the true state of the process"
- What do we need?
  - Data model (measurement model)

Introduction

- What is DA?
  - Various definitions
  - Their definition: "DA is an approach for fusing data (observations) with prior knowledge to obtain an estimate of the distribution of the true state of the process"
- What do we need?
  - Data model (measurement model)
  - Process model

Introduction

- What is DA?
  - Various definitions
  - Their definition: "DA is an approach for fusing data (observations) with prior knowledge to obtain an estimate of the distribution of the true state of the process"
- What do we need?
  - Data model (measurement model)
  - Process model
- Why do we need these models?

Introduction

- What is DA?
  - Various definitions
  - Their definition: "DA is an approach for fusing data (observations) with prior knowledge to obtain an estimate of the distribution of the true state of the process"
- What do we need?
  - Data model (measurement model)
  - Process model
- Why do we need these models?
  - We need them for *Bayesian Inference*

Bayesian Inference

- What is Bayesian Inference and how do we do it?

Bayesian Inference

- What is Bayesian Inference and how do we do it?
- There are 3 steps

Bayesian Inference

- What is Bayesian Inference and how do we do it?
- There are 3 steps
  1. Formulate ‘full’ probability model $p(x, y)$

Bayesian Inference

- What is Bayesian Inference and how do we do it?
- There are 3 steps
  1. Formulate ‘full’ probability model $p(x, y)$
  2. Find $p(x|y)$

Bayesian Inference

- What is Bayesian Inference and how do we do it?
- There are 3 steps
  1. Formulate ‘full’ probability model $p(x, y)$
  2. Find $p(x|y)$
  3. Evaluate fit and validity of model

Bayesian Inference

- What is Bayesian Inference and how do we do it?
- There are 3 steps
  1. Formulate 'full' probability model $p(x, y)$
  2. Find $p(x|y)$
  3. Evaluate fit and validity of model
- Conditional probability gives:

$$p(x, y) = p(x|y)p(y) = p(y|x)p(x)$$

Bayesian Inference

- What is Bayesian Inference and how do we do it?
- There are 3 steps
  1. Formulate 'full' probability model $p(x, y)$
  2. Find $p(x|y)$
  3. Evaluate fit and validity of model
- Conditional probability gives:

$$p(x, y) = p(x|y)p(y) = p(y|x)p(x)$$

- Using this we obtain Bayes' Rule:

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \quad \text{for } 0 < p(y) < \infty$$

Bayesian Inference

- $p(x)$ is our prior belief or *prior distribution*

Bayesian Inference

- $p(x)$ is our prior belief or *prior distribution*
- $p(y|x)$ is the data (or measurement) model. These are our observations

Bayesian Inference

- $p(x)$ is our prior belief or *prior distribution*
- $p(y|x)$ is the data (or measurement) model. These are our observations
- Using these and Bayes' Rule (on the previous slide) we can find $p(x|y)$, the state of the system *given* the observations

Example: Univariate Normal-Normal Case

- Let's say our prior knowledge is: $X \sim \mathcal{N}(\mu, \tau^2)$

Example: Univariate Normal-Normal Case

- Let's say our prior knowledge is: $X \sim \mathcal{N}(\mu, \tau^2)$
- And we have n independent observations, $Y = (Y_1, \dots, Y_n)^T$, where

$$Y_i | X \sim \mathcal{N}(x, \sigma^2)$$

Example: Univariate Normal-Normal Case

- Let's say our prior knowledge is: $X \sim \mathcal{N}(\mu, \tau^2)$
- And we have n independent observations, $Y = (Y_1, \ldots, Y_n)^T$, where

$$Y_i | X \sim \mathcal{N}(x, \sigma^2)$$

- Equation for the prior is:

$$p(x) = \frac{1}{\sqrt{2\pi\tau^2}} \exp\left(-\frac{1}{2\tau^2}(x-\mu)^2\right)$$

Example: Univariate Normal-Normal Case

- As a result of independent observations, we have

Example: Univariate Normal-Normal Case

- As a result of independent observations, we have

$$
\begin{aligned}
p(y|x) &= \prod_{i=1}^{n} \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right) \exp\left(-\frac{1}{2\sigma^2} (y_i - x)^2 \right) \\
&\propto \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - x)^2 \right)
\end{aligned}
$$

- We can work out the posterior distribution:

Example: Univariate Normal-Normal Case

- As a result of independent observations, we have

$$p(y|x) = \prod_{i=1}^{n} \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right) \exp\left(-\frac{1}{2\sigma^2}(y_i - x)^2 \right)$$

$$\propto \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - x)^2 \right)$$

- We can work out the posterior distribution:

$$p(x|y) \propto \exp\left(-\frac{1}{2} \left[\sum_{i=1}^{n} \frac{(y_i - x)^2}{\sigma^2} + \frac{(x-\mu)^2}{\tau^2} \right] \right)$$

$$\propto \exp\left(-\frac{1}{2} \left[\sum_{i=1}^{n} \frac{-2xy_i + x^2}{\sigma^2} + \frac{x^2 - 2x\mu}{\tau^2} \right] \right)$$

Example: Univariate Normal-Normal Case

$$= \exp\left(-\frac{1}{2}\left(\frac{n}{\sigma^2} + \frac{1}{\tau^2}\right)x^2 - 2\left(\sum_{i=1}^{n}\frac{y_i}{\sigma^2} + \frac{\mu}{\tau^2}\right)x\right)$$

$$= \exp\left(-\frac{1}{2}(ax^2 - 2bx)\right)$$

$$\propto \exp\left(-\frac{a}{2}\left(x - \frac{b}{a}\right)^2\right)$$

Example: Univariate Normal-Normal Case

$$
\begin{aligned}
&= \exp\left(-\frac{1}{2}\left(\frac{n}{\sigma^2}+\frac{1}{\tau^2}\right)x^2 - 2\left(\sum_{i=1}^{n}\frac{y_i}{\sigma^2}+\frac{\mu}{\tau^2}\right)x\right) \\
&= \exp\left(-\frac{1}{2}(ax^2-2bx)\right) \\
&\propto \exp\left(-\frac{a}{2}\left(x-\frac{b}{a}\right)^2\right)
\end{aligned}
$$

- So we have:

$$
p(x|y) = \mathcal{N}\left(\left(\frac{n}{\sigma^2}+\frac{1}{\tau^2}\right)^{-1}\left(\sum_{i=1}^{n}\frac{y_i}{\sigma^2}+\frac{\mu}{\tau^2}\right), \left(\frac{n}{\sigma^2}+\frac{1}{\tau^2}\right)^{-1}\right)
$$

Example: Univariate Normal-Normal Case

- Write

Example: Univariate Normal-Normal Case

- Write

$$\bar{y} = \sum_{i=1}^{n} y_i / n$$

$$\omega_y = \frac{n\tau^2}{n\tau^2 + \sigma^2}$$

$$\omega_\mu = \frac{\sigma^2}{n\tau^2 + \sigma^2}$$

Example: Univariate Normal-Normal Case

- Write

$$\bar{y} = \sum_{i=1}^{n} y_i / n$$

$$\omega_y = \frac{n\tau^2}{n\tau^2 + \sigma^2}$$

$$\omega_\mu = \frac{\sigma^2}{n\tau^2 + \sigma^2}$$

- Then we have

$$\mathbb{E}(X|y) = \omega_y \bar{y} + \omega_\mu \mu$$

Example: Univariate Normal-Normal Case

- τ^2 or $n \to \infty \Rightarrow \omega_y \to 1$ and $\omega_\mu \to 0$ (so data dominates prior)

Example: Univariate Normal-Normal Case

- τ^2 or $n \to \infty \Rightarrow \omega_y \to 1$ and $\omega_\mu \to 0$ (so data dominates prior)
- For small τ^2, the prior is critical for small n ($\omega_y \to 0$)

Example: Univariate Normal-Normal Case

- τ^2 or $n \to \infty \Rightarrow \omega_y \to 1$ and $\omega_\mu \to 0$ (so data dominates prior)
- For small τ^2, the prior is critical for small n ($\omega_y \to 0$)
- Also, can write

$$\begin{aligned}\mathbb{E}(X|y) &= \mu + \omega_y(\bar{y} - \mu) \\ &= \mu + K(\bar{y} - \mu)\end{aligned}$$

Gain K adjusts prior mean towards sample mean.

Example: Univariate Normal-Normal Case

- τ^2 or $n \to \infty \Rightarrow \omega_y \to 1$ and $\omega_\mu \to 0$ (so data dominates prior)
- For small τ^2, the prior is critical for small n ($\omega_y \to 0$)
- Also, can write

$$\begin{aligned} \mathbb{E}(X|y) &= \mu + \omega_y(\bar{y} - \mu) \\ &= \mu + K(\bar{y} - \mu) \end{aligned}$$

  Gain K adjusts prior mean towards sample mean.

- Similar for variance

$$\mathrm{Var}(X|y) = (1 - K)\tau^2$$

Sequential Approaches

- Idea: update posterior sequentially

Sequential Approaches

- Idea: update posterior sequentially
- Notation

$$Y_{1:t} = \{Y_1, \ldots, Y_t\}, \quad X_{0:t} = \{X_0, \ldots, X_t\}$$

Sequential Approaches

- Idea: update posterior sequentially
- Notation

$$Y_{1:t} = \{Y_1, \dots, Y_t\}, \quad X_{0:t} = \{X_0, \dots, X_t\}$$

- Assumptions:

$$p(x_{0:T}) = p(x_0) \prod_{t=1}^{T} p(x_t | x_{t-1})$$

$$p(y_{1:T} | x_{0:T}) = \prod_{i=1}^{T} p(y_t | x_t)$$

Sequential Approaches

- Idea: update posterior sequentially
- Notation

$$Y_{1:t} = \{Y_1, \ldots, Y_t\}, \quad X_{0:t} = \{X_0, \ldots, X_t\}$$

- Assumptions:

$$p(x_{0:T}) = p(x_0) \prod_{t=1}^{T} p(x_t|x_{t-1})$$

$$p(y_{1:T}|x_{0:T}) = \prod_{i=1}^{T} p(y_t|x_t)$$

- Using the above we have:

$$p(x_{0:T}|y_{1:T}) \propto p(x_0) \prod_{t=1}^{T} p(y_t|x_t) p(x_t|x_{t-1})$$

Filtering

- Filtering: use only past data to update posterior

Filtering

- Filtering: use only past data to update posterior
- Given $p(x_{t-1}|y_{1:t-1})$, find forecast $p(x_t|y_{1:t-1})$ and analysis $p(x_t|y_{1:t})$

Filtering

- Filtering: use only past data to update posterior
- Given $p(x_{t-1}|y_{1:t-1})$, find forecast $p(x_t|y_{1:t-1})$ and analysis $p(x_t|y_{1:t})$
- Markovian assumption gives:

$$p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})\,dx_{t-1}$$

Filtering

- Filtering: use only past data to update posterior
- Given $p(x_{t-1}|y_{1:t-1})$, find forecast $p(x_t|y_{1:t-1})$ and analysis $p(x_t|y_{1:t})$
- Markovian assumption gives:

$$p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})\, dx_{t-1}$$

- Bayes' Rule gives:

$$\begin{aligned} p(x_t|y_{1:t}) &= p(x_t|y_t, y_{1:t-1}) \\ &\propto p(y_t|x_t, y_{1:t-1})p(x_t|y_{1:t-1}) \\ &= p(y_t|x_t)p(x_t|y_{1:t-1}) \end{aligned}$$

Filtering

- Filtering: use only past data to update posterior
- Given $p(x_{t-1}|y_{1:t-1})$, find forecast $p(x_t|y_{1:t-1})$ and analysis $p(x_t|y_{1:t})$
- Markovian assumption gives:

$$p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})\, dx_{t-1}$$

- Bayes' Rule gives:

$$\begin{aligned} p(x_t|y_{1:t}) &= p(x_t|y_t, y_{1:t-1}) \\ &\propto p(y_t|x_t, y_{1:t-1})p(x_t|y_{1:t-1}) \\ &= p(y_t|x_t)p(x_t|y_{1:t-1}) \end{aligned}$$

- So we can iterate between forecast and analysis distributions to update the posterior distribution

Smoothing

- Smoothing: use *all* the data to update posterior

Smoothing

- Smoothing: use *all* the data to update posterior
- Want $p(x_{0:T}|y_{1:T})$ in sequential fashion

Smoothing

- Smoothing: use *all* the data to update posterior
- Want $p(x_{0:T}|y_{1:T})$ in sequential fashion
- Smoothing distribution is given by:

$$p(x_t|y_{1:T}) = \int p(x_t|x_{t+1}, y_{1:T})p(x_{t+1}|y_{1:T})\, dx_{t+1}$$

Smoothing

- Smoothing: use *all* the data to update posterior
- Want $p(x_{0:T}|y_{1:T})$ in sequential fashion
- Smoothing distribution is given by:

$$p(x_t|y_{1:T}) = \int p(x_t|x_{t+1}, y_{1:T})p(x_{t+1}|y_{1:T})\, dx_{t+1}$$

- Assume: $p(x_t|x_{t+1}, y_{1:T}) = p(x_t|x_{t+1}, y_{1:t})$

Smoothing

- Smoothing: use *all* the data to update posterior
- Want $p(x_{0:T}|y_{1:T})$ in sequential fashion
- Smoothing distribution is given by:

$$p(x_t|y_{1:T}) = \int p(x_t|x_{t+1}, y_{1:T})p(x_{t+1}|y_{1:T})\, dx_{t+1}$$

- Assume: $p(x_t|x_{t+1}, y_{1:T}) = p(x_t|x_{t+1}, y_{1:t})$
- Bayes' Rule gives:

$$\begin{aligned} p(x_t|x_{t+1}, y_{1:t}) &\propto p(x_{t+1}|x_t, y_{1:t})p(x_t|y_{1:t}) \\ &= p(x_{t+1}|x_t)p(x_t|y_{1:t}) \end{aligned}$$

Forward Filtering-Backward Smoothing Algorithm

- Forward Filtering-Backward Smoothing Algorithm

Forward Filtering-Backward Smoothing Algorithm

- Forward Filtering-Backward Smoothing Algorithm
- For $t = T - 1$ to 1

Forward Filtering-Backward Smoothing Algorithm

- Forward Filtering-Backward Smoothing Algorithm
- For $t = T - 1$ to 1
  1. multiply filtered analysis distribution $p(x_t|y_{1:t})$ with $p(x_{t+1}|x_t)$ to obtain $p(x_t|x_{t+1}, y_{1:t})$

Forward Filtering-Backward Smoothing Algorithm

- Forward Filtering-Backward Smoothing Algorithm
- For $t = T - 1$ to 1
  1. multiply filtered analysis distribution $p(x_t|y_{1:t})$ with $p(x_{t+1}|x_t)$ to obtain $p(x_t|x_{t+1}, y_{1:t})$
  2. use smoothing distribution definition to obtain the smoothing distribution $p(x_t|y_{1:T})$ making use of smoothing distribution at time $t+1$, $p(x_{t+1}|y_{1:T})$, obtained at previous iteration

Forward Filtering-Backward Smoothing Algorithm

- Forward Filtering-Backward Smoothing Algorithm
- For $t = T - 1$ to 1
  1. multiply filtered analysis distribution $p(x_t | y_{1:t})$ with $p(x_{t+1} | x_t)$ to obtain $p(x_t | x_{t+1}, y_{1:t})$
  2. use smoothing distribution definition to obtain the smoothing distribution $p(x_t | y_{1:T})$ making use of smoothing distribution at time $t + 1$, $p(x_{t+1} | y_{1:T})$, obtained at previous iteration
- Aside: For a linear model, linear observation operator and Gaussian prior, the posterior is Gaussian. Its mean and covariance can be updated iteratively in a similar manner. This is known as the Kalman Filter and Kalman Smoother.

Basic Monte Carlo Sampling

- What if the model (or observation operator) is nonlinear?

Basic Monte Carlo Sampling

- What if the model (or observation operator) is nonlinear?
  - Linearise locally (Extended Kalman Filter)

Basic Monte Carlo Sampling

- What if the model (or observation operator) is nonlinear?
  - Linearise locally (Extended Kalman Filter)
  - Use Monte Carlo sampling

Basic Monte Carlo Sampling

- What if the model (or observation operator) is nonlinear?
  - Linearise locally (Extended Kalman Filter)
  - Use Monte Carlo sampling
- Basic MC is to estimate integrals. Say we want to calculate:

$$\begin{aligned}
\mathbb{E}(f(X_{0:t})|y_{1:t}) &= \int f(x_{0:t})p(x_{0:t}|y_{1:t})\,dx_{0:t} \\
&= \frac{\int f(x_{0:t})p(y_{1:t}|x_{0:t})p(x_{0:t})\,dx_{0:t}}{\int p(y_{1:t}|x_{0:t})p(x_{0:t})\,dx_{0:t}}
\end{aligned}$$

Basic Monte Carlo Sampling

- What if the model (or observation operator) is nonlinear?
    - Linearise locally (Extended Kalman Filter)
    - Use Monte Carlo sampling
- Basic MC is to estimate integrals. Say we want to calculate:

$$\begin{aligned}\mathbb{E}(f(X_{0:t})|y_{1:t}) &= \int f(x_{0:t})p(x_{0:t}|y_{1:t})\,dx_{0:t} \\ &= \frac{\int f(x_{0:t})p(y_{1:t}|x_{0:t})p(x_{0:t})\,dx_{0:t}}{\int p(y_{1:t}|x_{0:t})p(x_{0:t})\,dx_{0:t}}\end{aligned}$$

- We can use m pseudo-random realisations, $x_{0:t}^{i}$, from $p(x_{0:t}|y_{1:t})$ and compute:

$$\hat{\mathbb{E}}(f(X_{0:t})|y_{1:t}) = \frac{1}{M}\sum_{i=1}^{M} f(x_{0:t}^{i})$$

Basic Monte Carlo Sampling

- What if the model (or observation operator) is nonlinear?
  - Linearise locally (Extended Kalman Filter)
  - Use Monte Carlo sampling
- Basic MC is to estimate integrals. Say we want to calculate:

$$
\begin{aligned}
\mathbb{E}(f(X_{0:t})|y_{1:t}) &= \int f(x_{0:t})p(x_{0:t}|y_{1:t})\, dx_{0:t} \\
&= \frac{\int f(x_{0:t})p(y_{1:t}|x_{0:t})p(x_{0:t})\, dx_{0:t}}{\int p(y_{1:t}|x_{0:t})p(x_{0:t})\, dx_{0:t}}
\end{aligned}
$$

- We can use m pseudo-random realisations, $x^i_{0:t}$, from $p(x_{0:t}|y_{1:t})$ and compute:

$$
\hat{\mathbb{E}}(f(X_{0:t})|y_{1:t}) = \frac{1}{M}\sum_{i=1}^{M} f(x^i_{0:t})
$$

- Can approximate $p(x_{0:t}|y_{1:t})$ by $p^m(x_{0:t}|y_{1:t}) = \sum_{i=1}^{M} \delta_{x^i_{0:t}}$

Importance Sampling Monte Carlo

- Use ISMC when drawing from $p(x_{0:t}|y_{1:t})$ is hard

Importance Sampling Monte Carlo

- Use ISMC when drawing from $p(x_{0:t}|y_{1:t})$ is hard
- Sample from $g(x_{0:t}|y_{1:t}) \gg p(x_{0:t}|y_{1:t})$, but weight each sample and calculate:

$$\hat{\mathbb{E}}(f(X_{0:t})|y_{1:t}) = \sum_{i=1}^{M} w^i f(x_{0:t}^i)$$

Importance Sampling Monte Carlo

- Use ISMC when drawing from $p(x_{0:t}|y_{1:t})$ is hard
- Sample from $g(x_{0:t}|y_{1:t}) \gg p(x_{0:t}|y_{1:t})$, but weight each sample and calculate:

$$\hat{\mathbb{E}}(f(X_{0:t})|y_{1:t}) = \sum_{i=1}^{M} w^i f(x^i_{0:t})$$

where

$$w^i = \frac{p(x^i_{0:t}|y_{1:t})/g(x^i_{0:t}|y_{1:t})}{\sum_{j=1}^{M} p(x^j_{0:t}|y_{1:t})/g(x^j_{0:t}|y_{1:t})}$$

Importance Sampling Monte Carlo

- Use ISMC when drawing from $p(x_{0:t}|y_{1:t})$ is hard
- Sample from $g(x_{0:t}|y_{1:t}) \gg p(x_{0:t}|y_{1:t})$, but weight each sample and calculate:

$$\hat{\mathbb{E}}(f(X_{0:t})|y_{1:t}) = \sum_{i=1}^{M} w^i f(x_{0:t}^i)$$

where

$$w^i = \frac{p(x_{0:t}^i|y_{1:t})/g(x_{0:t}^i|y_{1:t})}{\sum_{j=1}^{M} p(x_{0:t}^j|y_{1:t})/g(x_{0:t}^j|y_{1:t})}$$

- Can approximate $p(x_{0:t}|y_{1:t})$ by $p^m(x_{0:t}|y_{1:t}) = \sum_{i=1}^{M} w^i \delta_{x_{0:t}^i}$

Importance Sampling Monte Carlo

- Usually choose $g(x_{0:t}|y_{1:t}) = p(x_{0:t})$

Importance Sampling Monte Carlo

- Usually choose $g(x_{0:t}|y_{1:t}) = p(x_{0:t})$
- Given a sample from $p(x_{0:t})$, simulate the forward model to obtain Monte Carlo trajectories

Importance Sampling Monte Carlo

- Usually choose $g(x_{0:t}|y_{1:t}) = p(x_{0:t})$
- Given a sample from $p(x_{0:t})$, simulate the forward model to obtain Monte Carlo trajectories
- With this g, we get:

$$\begin{aligned} w^i &\propto \frac{p(x_{0:t}^i|y_{1:t})}{p(x_{0:t}^i} \\ &\propto p(y_{1:t}|x_{0:t}^i) \end{aligned}$$

Importance Sampling Monte Carlo

- Usually choose $g(x_{0:t}|y_{1:t}) = p(x_{0:t})$
- Given a sample from $p(x_{0:t})$, simulate the forward model to obtain Monte Carlo trajectories
- With this g, we get:

$$
\begin{aligned}
w^i &\propto \frac{p(x^i_{0:t}|y_{1:t})}{p(x^i_{0:t}} \\
&\propto p(y_{1:t}|x^i_{0:t})
\end{aligned}
$$

- This is called an *ensemble smoother*

Sequential Monte Carlo

- Sequential MC algorithm follows from Filtering algorithm

Sequential Monte Carlo

- Sequential MC algorithm follows from Filtering algorithm
- Have seen: $g = prior \Rightarrow w^i \propto likelihood$

Sequential Monte Carlo

- Sequential MC algorithm follows from Filtering algorithm
- Have seen: $g = prior \Rightarrow w^i \propto likelihood$
- Let w_t^i be weight on ensemble member i at time t

Sequential Monte Carlo

- Sequential MC algorithm follows from Filtering algorithm
- Have seen: $g = prior \Rightarrow w^i \propto likelihood$
- Let w_t^i be weight on ensemble member i at time t
- Independent data gives:

$$w_t^i \propto p(y_{1:t}|x_{0:t}^i) \propto p(y_t|x_t^i)w_{t-1}^i$$

Sequential Monte Carlo

- Sequential MC algorithm follows from Filtering algorithm
- Have seen: $g = \mathit{prior} \Rightarrow w^i \propto \mathit{likelihood}$
- Let w_t^i be weight on ensemble member i at time t
- Independent data gives:

$$w_t^i \propto p(y_{1:t}|x_{0:t}^i) \propto p(y_t|x_t^i) w_{t-1}^i$$

- Approximate analysis distribution at time $t-1$ by

$$p^m(x_{t-1}|y_{1:t-1}) = \sum_{i=1}^{M} w_{t-1}^i \delta_{x_{t-1|t-1}^i}$$

Sequential Monte Carlo

- Sequential MC algorithm follows from Filtering algorithm
- Have seen: $g = prior \Rightarrow w^i \propto likelihood$
- Let w_t^i be weight on ensemble member i at time t
- Independent data gives:

$$w_t^i \propto p(y_{1:t}|x_{0:t}^i) \propto p(y_t|x_t^i)w_{t-1}^i$$

- Approximate analysis distribution at time $t-1$ by

$$p^m(x_{t-1}|y_{1:t-1}) = \sum_{i=1}^{M} w_{t-1}^i \delta_{x_{t-1|t-1}^i}$$

where $x_{t-1|t-1}^i$ are random draws from $p(x_{t-1}|y_{1:t-1})$

Sequential Monte Carlo

- In practice one considers a kernel-density approximation of this and so we can estimate the forecast:

Sequential Monte Carlo

- In practice one considers a kernel-density approximation of this and so we can estimate the forecast:

$$p^m(x_t|y_{1:t-1}) \propto \sum_{i=1}^{M} p(x_t|x_{t-1:t-1}^i)w_{t-1}^i$$

Sequential Monte Carlo

- In practice one considers a kernel-density approximation of this and so we can estimate the forecast:

$$p^m(x_t|y_{1:t-1}) \propto \sum_{i=1}^{M} p(x_t|x^i_{t-1:t-1})w^i_{t-1}$$

- Using the weight-update above we have:

$$\begin{aligned} p^m(x_t|y_{1:1}) &\propto p(y_t|x_t)\sum_{i=1}^{M} p(x_t|x^i_{t-1:t-1})w^i_{t-1} \\ &= \sum_{i=1}^{M} p(x_t|x^i_{t-1:t-1})w^i_t \end{aligned}$$

Sequential Monte Carlo

- In practice one considers a kernel-density approximation of this and so we can estimate the forecast:

$$p^m(x_t|y_{1:t-1}) \propto \sum_{i=1}^{M} p(x_t|x^i_{t-1:t-1})w^i_{t-1}$$

- Using the weight-update above we have:

$$\begin{aligned} p^m(x_t|y_{1:1}) &\propto p(y_t|x_t)\sum_{i=1}^{M} p(x_t|x^i_{t-1:t-1})w^i_{t-1} \\ &= \sum_{i=1}^{M} p(x_t|x^i_{t-1:t-1})w^i_t \end{aligned}$$

- Various methods to resolve problems from dimensionality...

Particle Filtering

- Throw away samples with small weight

Particle Filtering

- Throw away samples with small weight
- Multiply particles with high weight

Particle Filtering

- Throw away samples with small weight
- Multiply particles with high weight
- Use Bootstrap Algorithm:

Particle Filtering

- Throw away samples with small weight
- Multiply particles with high weight
- Use Bootstrap Algorithm:
    1. Initialisation, $t = 0$

Particle Filtering

- Throw away samples with small weight
- Multiply particles with high weight
- Use Bootstrap Algorithm:
  1. Initialisation, $t = 0$
     - for $i = 0, \dots, m$ sample $x^i_{0|0} \sim p(x_0)$ and set $t = 1$

Particle Filtering

- Throw away samples with small weight
- Multiply particles with high weight
- Use Bootstrap Algorithm:
  1. Initialisation, $t = 0$
     - for $i = 0, \dots, m$ sample $x^i_{0|0} \sim p(x_0)$ and set $t = 1$
  2. Importance sampling step

Particle Filtering

- Throw away samples with small weight
- Multiply particles with high weight
- Use Bootstrap Algorithm:
  1. Initialisation, $t = 0$
     - for $i = 0, \ldots, m$ sample $x^i_{0|0} \sim p(x_0)$ and set $t = 1$
  2. Importance sampling step
     - for $i = 0, \ldots, m$ sample $\tilde{x}^i_t \sim p(x_t | x^i_{t-1})$ and set $\tilde{x}^i_t = \{x^i_{t-1}, \tilde{x}^i_t\}$

Particle Filtering

- Throw away samples with small weight
- Multiply particles with high weight
- Use Bootstrap Algorithm:
  1. Initialisation, $t = 0$
     - for $i = 0, \ldots, m$ sample $x^i_{0|0} \sim p(x_0)$ and set $t = 1$
  2. Importance sampling step
     - for $i = 0, \ldots, m$ sample $\tilde{x}^i_t \sim p(x_t | x^i_{t-1})$ and set $\tilde{x}^i_t = \{x^i_{t-1}, \tilde{x}^i_t\}$
     - for $i = 0, \ldots, m$ evaluate importance weights $\tilde{w}^i_t = p(y_t | \tilde{x}^i_t)$

Particle Filtering

- Throw away samples with small weight
- Multiply particles with high weight
- Use Bootstrap Algorithm:
  1. Initialisation, $t = 0$
     - for $i = 0, \dots, m$ sample $x^i_{0|0} \sim p(x_0)$ and set $t = 1$
  2. Importance sampling step
     - for $i = 0, \dots, m$ sample $\tilde{x}^i_t \sim p(x_t | x^i_{t-1})$ and set $\tilde{x}^i_t = \{x^i_{t-1}, \tilde{x}^i_t\}$
     - for $i = 0, \dots, m$ evaluate importance weights $\tilde{w}^i_t = p(y_t | \tilde{x}^i_t)$
     - normalise weights

Particle Filtering

- Throw away samples with small weight
- Multiply particles with high weight
- Use Bootstrap Algorithm:
  1. Initialisation, $t = 0$
     - for $i = 0, \ldots, m$ sample $x^i_{0|0} \sim p(x_0)$ and set $t = 1$
  2. Importance sampling step
     - for $i = 0, \ldots, m$ sample $\tilde{x}^i_t \sim p(x_t | x^i_{t-1})$ and set $\tilde{x}^i_t = \{x^i_{t-1}, \tilde{x}^i_t\}$
     - for $i = 0, \ldots, m$ evaluate importance weights $\tilde{w}^i_t = p(y_t | \tilde{x}^i_t)$
     - normalise weights
  3. Selection step

Particle Filtering

- Throw away samples with small weight
- Multiply particles with high weight
- Use Bootstrap Algorithm:
  1. Initialisation, $t = 0$
     - for $i = 0, \dots, m$ sample $x^i_{0|0} \sim p(x_0)$ and set $t = 1$
  2. Importance sampling step
     - for $i = 0, \dots, m$ sample $\tilde{x}^i_t \sim p(x_t | x^i_{t-1})$ and set $\tilde{x}^i_t = \{x^i_{t-1}, \tilde{x}^i_t\}$
     - for $i = 0, \dots, m$ evaluate importance weights $\tilde{w}^i_t = p(y_t | \tilde{x}^i_t)$
     - normalise weights
  3. Selection step
     - resample with replacement m particles $\{x^i_t : i = 1, \dots, m\}$ from the set $\{\tilde{x}^i_t : i = 1, \dots, m\}$ according to importance weights

Particle Filtering

- Throw away samples with small weight
- Multiply particles with high weight
- Use Bootstrap Algorithm:
  1. Initialisation, $t = 0$
     - for $i = 0, \dots, m$ sample $x^i_{0|0} \sim p(x_0)$ and set $t = 1$
  2. Importance sampling step
     - for $i = 0, \dots, m$ sample $\tilde{x}^i_t \sim p(x_t | x^i_{t-1})$ and set $\tilde{x}^i_t = \{x^i_{t-1}, \tilde{x}^i_t\}$
     - for $i = 0, \dots, m$ evaluate importance weights $\tilde{w}^i_t = p(y_t | \tilde{x}^i_t)$
     - normalise weights
  3. Selection step
     - resample with replacement m particles $\{x^i_t : i = 1, \dots, m\}$ from the set $\{\tilde{x}^i_t : i = 1, \dots, m\}$ according to importance weights
     - set $t = t + 1$ and go to step 2

Ensemble Kalman Filter (EnKF)

- Can use the Ensmeble Kalman Filter in high dimensional problems

Ensemble Kalman Filter (EnKF)

- Can use the Ensmeble Kalman Filter in high dimensional problems
- Approach:

Ensemble Kalman Filter (EnKF)

- Can use the Ensmeble Kalman Filter in high dimensional problems
- Approach:
  - Use MC samples to approximate forecast mean and covariance while still using nonlinear forward model

Ensemble Kalman Filter (EnKF)

- Can use the Ensmeble Kalman Filter in high dimensional problems
- Approach:
  - Use MC samples to approximate forecast mean and covariance while still using nonlinear forward model
  - Use the these in linear Kalman Filter update formulas to obtain analysis distribution

Ensemble Kalman Filter (EnKF)

- Can use the Ensmeble Kalman Filter in high dimensional problems
- Approach:
  - Use MC samples to approximate forecast mean and covariance while still using nonlinear forward model
  - Use the these in linear Kalman Filter update formulas to obtain analysis distribution
- Assume we have m independent samples from analysis distribution at time $t-1$, $x^i_{t-1|t-1}$.

Ensemble Kalman Filter (EnKF)

- Can use the Ensmeble Kalman Filter in high dimensional problems
- Approach:
  - Use MC samples to approximate forecast mean and covariance while still using nonlinear forward model
  - Use the these in linear Kalman Filter update formulas to obtain analysis distribution
- Assume we have m independent samples from analysis distribution at time $t-1$, $x^i_{t-1|t-1}$.
- Also assume $w^i_{t-1} = 1/m$

Ensemble Kalman Filter (EnKF)

- Can use the Ensmeble Kalman Filter in high dimensional problems
- Approach:
    - Use MC samples to approximate forecast mean and covariance while still using nonlinear forward model
    - Use the these in linear Kalman Filter update formulas to obtain analysis distribution
- Assume we have m independent samples from analysis distribution at time $t-1$, $x^i_{t-1|t-1}$.
- Also assume $w^i_{t-1} = 1/m$
- Use these samples and basic MC to approximate forecast:

$$p^m(x_t|y_{1:t-1}) = (1/m) \sum_{i=1}^{M} p(x_t|x^i_{t-1:t-1})$$

Ensemble Kalman Filter (EnKF)

- Now assume forecast distribution can be characterised by first two moments (or Guassian) with mean $x^i_{t|t-1}$ and estimated covariance matrix $\hat{P}_{t|t-1}$. Then analysis (update) distribution is given by:

Ensemble Kalman Filter (EnKF)

- Now assume forecast distribution can be characterised by first two moments (or Guassian) with mean $x^i_{t|t-1}$ and estimated covariance matrix $\hat{P}_{t|t-1}$. Then analysis (update) distribution is given by:

$$p^m(x_t|y_{1:t}) \propto (1/m)p(y_t|x_t)\sum_{i=1}^{M}\mathcal{N}(x^i_{t|t-1}, \hat{P}_{t|t-1})$$

Ensemble Kalman Filter (EnKF)

- Now assume forecast distribution can be characterised by first two moments (or Guassian) with mean $x^i_{t|t-1}$ and estimated covariance matrix $\hat{P}_{t|t-1}$. Then analysis (update) distribution is given by:

$$p^m(x_t|y_{1:t}) \propto (1/m)p(y_t|x_t)\sum_{i=1}^{M}\mathcal{N}(x^i_{t|t-1}, \hat{P}_{t|t-1})$$

- Assume a linear (or linearised) observation model:

$$y_t = H_t x_t + e_t$$

Ensemble Kalman Filter (EnKF)

- Now assume forecast distribution can be characterised by first two moments (or Guassian) with mean $x^i_{t|t-1}$ and estimated covariance matrix $\hat{P}_{t|t-1}$. Then analysis (update) distribution is given by:

$$p^m(x_t|y_{1:t}) \propto (1/m)p(y_t|x_t)\sum_{i=1}^{M}\mathcal{N}(x^i_{t|t-1}, \hat{P}_{t|t-1})$$

- Assume a linear (or linearised) observation model:

$$y_t = H_t x_t + e_t$$

where $\text{Cov}(e_t) = R_t$

Ensemble Kalman Filter (EnKF)

- Now assume forecast distribution can be characterised by first two moments (or Guassian) with mean $x^i_{t|t-1}$ and estimated covariance matrix $\hat{P}_{t|t-1}$. Then analysis (update) distribution is given by:

$$p^m(x_t|y_{1:t}) \propto (1/m)p(y_t|x_t)\sum_{i=1}^{M}\mathcal{N}(x^i_{t|t-1}, \hat{P}_{t|t-1})$$

- Assume a linear (or linearised) observation model:

$$y_t = H_t x_t + e_t$$

  where $\text{Cov}(e_t) = R_t$

- Let $x_{t|t-1}$ and $P_{t|t-1}$ be mean and covariance of $p(x_t|y_{1:t-1})$

Ensemble Kalman Filter (EnKF)

- Now assume forecast distribution can be characterised by first two moments (or Guassian) with mean $x^i_{t|t-1}$ and estimated covariance matrix $\hat{P}_{t|t-1}$. Then analysis (update) distribution is given by:

$$p^m(x_t|y_{1:t}) \propto (1/m)p(y_t|x_t)\sum_{i=1}^{M}\mathcal{N}(x^i_{t|t-1}, \hat{P}_{t|t-1})$$

- Assume a linear (or linearised) observation model:

$$y_t = H_t x_t + e_t$$

where $\mathrm{Cov}(e_t) = R_t$

- Let $x_{t|t-1}$ and $P_{t|t-1}$ be mean and covariance of $p(x_t|y_{1:t-1})$
- Want a viable ensemble from $p(x_t|y_{1:t})$, or of $x_{t|t}$ and $P_{t|t}$

Ensemble Kalman Filter (EnKF)

Algorithm:

Ensemble Kalman Filter (EnKF)

Algorithm:

- Evolve $x^i_{t-1|t-1}$ forward using:

$$x^i_{t|t-1} = \mathcal{M}(x^i_{t-1|t-1}) + \eta^i_t, \quad \eta^i_t \sim \mathcal{N}(0, Q)$$

Ensemble Kalman Filter (EnKF)

Algorithm:

- Evolve $x^i_{t-1|t-1}$ forward using:

$$x^i_{t|t-1} = \mathcal{M}(x^i_{t-1|t-1}) + \eta^i_t, \quad \eta^i_t \sim \mathcal{N}(0, Q)$$

- Use evolved samples to calculate a sample forecast covariance matrix, $\hat{P}_{t|t-1}$:

$$\hat{P}_{t|t-1} = \frac{1}{m-1} \sum_{i=1}^{m} (x^i_{t|t-1} - \hat{x}^i_{t|t-1})(x^i_{t|t-1} - \hat{x}^i_{t|t-1})^T$$

Ensemble Kalman Filter (EnKF)

Algorithm:

- Evolve $x^i_{t-1|t-1}$ forward using:

$$x^i_{t|t-1} = \mathcal{M}(x^i_{t-1|t-1}) + \eta^i_t, \quad \eta^i_t \sim \mathcal{N}(0, Q)$$

- Use evolved samples to calculate a sample forecast covariance matrix, $\hat{P}_{t|t-1}$:

$$\hat{P}_{t|t-1} = \frac{1}{m-1} \sum_{i=1}^{m} (x^i_{t|t-1} - \hat{x}^i_{t|t-1})(x^i_{t|t-1} - \hat{x}^i_{t|t-1})^T$$

where $\hat{x}^i_{t|t-1} = (1/m) \sum_{i=1}^{m} x^i_{t|t-1}$

Ensemble Kalman Filter (EnKF)

Algorithm:

- Evolve $x^i_{t-1|t-1}$ forward using:

$$x^i_{t|t-1} = \mathcal{M}(x^i_{t-1|t-1}) + \eta^i_t, \quad \eta^i_t \sim \mathcal{N}(0, Q)$$

- Use evolved samples to calculate a sample forecast covariance matrix, $\hat{P}_{t|t-1}$:

$$\hat{P}_{t|t-1} = \frac{1}{m-1} \sum_{i=1}^{m} (x^i_{t|t-1} - \hat{x}^i_{t|t-1})(x^i_{t|t-1} - \hat{x}^i_{t|t-1})^T$$

  where $\hat{x}^i_{t|t-1} = (1/m) \sum_{i=1}^{m} x^i_{t|t-1}$

- Use Kalman update equations to update each forecast sample given the sampled observations:

$$x^i_{t|t} = x^i_{t|t-1} + K_t(y_t + e_t - H_t x^i_{t|t-1})$$

Ensemble Kalman Filter (EnKF)

Algorithm:

- Evolve $x^i_{t-1|t-1}$ forward using:

$$x^i_{t|t-1} = \mathcal{M}(x^i_{t-1|t-1}) + \eta^i_t, \quad \eta^i_t \sim \mathcal{N}(0, Q)$$

- Use evolved samples to calculate a sample forecast covariance matrix, $\hat{P}_{t|t-1}$:

$$\hat{P}_{t|t-1} = \frac{1}{m-1} \sum_{i=1}^{m} (x^i_{t|t-1} - \hat{x}^i_{t|t-1})(x^i_{t|t-1} - \hat{x}^i_{t|t-1})^T$$

where $\hat{x}^i_{t|t-1} = (1/m) \sum_{i=1}^{m} x^i_{t|t-1}$

- Use Kalman update equations to update each forecast sample given the sampled observations:

$$x^i_{t|t} = x^i_{t|t-1} + K_t(y_t + e_t - H_t x^i_{t|t-1})$$

where $K_t = \hat{P}_{t|t-1} H_t^T (H_t \hat{P}_{t|t-1} H_t^T + R)^{-1}$ and $e^i_t \sim \mathcal{N}(0, R)$

Summary

We have seen:

- How to use Bayes' Rule to infer on posterior

Summary

We have seen:

- How to use Bayes' Rule to infer on posterior
- How to update posterior sequentially (and analytically):

Summary

We have seen:

- How to use Bayes' Rule to infer on posterior
- How to update posterior sequentially (and analytically):
  - Filtering

Summary

We have seen:

- How to use Bayes' Rule to infer on posterior
- How to update posterior sequentially (and analytically):
  - Filtering
  - Smoothing

Summary

We have seen:

- How to use Bayes' Rule to infer on posterior
- How to update posterior sequentially (and analytically):
  - Filtering
  - Smoothing
- Forward-Filtering Backward-Smoothing

Summary

We have seen:

- How to use Bayes' Rule to infer on posterior
- How to update posterior sequentially (and analytically):
  - Filtering
  - Smoothing
- Forward-Filtering Backward-Smoothing
- Basic and Importance Monte Carlo sampling:

Summary

We have seen:

- How to use Bayes' Rule to infer on posterior
- How to update posterior sequentially (and analytically):
  - Filtering
  - Smoothing
- Forward-Filtering Backward-Smoothing
- Basic and Importance Monte Carlo sampling:
  - Sequential Monte Carlo

Summary

We have seen:

- How to use Bayes' Rule to infer on posterior
- How to update posterior sequentially (and analytically):
  - Filtering
  - Smoothing
- Forward-Filtering Backward-Smoothing
- Basic and Importance Monte Carlo sampling:
  - Sequential Monte Carlo
  - Bootstrap Algorithm

Summary

We have seen:

- How to use Bayes' Rule to infer on posterior
- How to update posterior sequentially (and analytically):
  - Filtering
  - Smoothing
- Forward-Filtering Backward-Smoothing
- Basic and Importance Monte Carlo sampling:
  - Sequential Monte Carlo
  - Bootstrap Algorithm
  - EnKF

Summary

We have seen:

- How to use Bayes' Rule to infer on posterior
- How to update posterior sequentially (and analytically):
  - Filtering
  - Smoothing
- Forward-Filtering Backward-Smoothing
- Basic and Importance Monte Carlo sampling:
  - Sequential Monte Carlo
  - Bootstrap Algorithm
  - EnKF
- Problems with these methods