Human Motion Analysis
Lecture 6: Bayesian Filtering

Raquel Urtasun

TTI Chicago

March 29, 2010

Materials used for this lecture

- This lecture is based on Zhe Chen's paper "Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond".
- I would like to thank David Fleet for his slides on the subject.
- To know more about sampling look at David MaKay's book "Information Theory, Inference, and Learning Algorithms", Cambridge University Press (2003).

Contents of today's lecture?

We will look into

- Stochastic Filtering Theory: Kalman filtering (1940's by Wiener and Kolmogorov).
- Bayesian Theory and Bayesian Filtering (Bayes, 1763 and rediscover by Laplace)
- Monte Carlo methods and Monte Carlo Filtering (Buffon 1777, modern version in the 1940's in physics and 1950's in statistics)

Monte Carlo approaches

- **Monte Carlo techniques** are stochastic sampling approaches aiming to tackle complex systems that are analytically intractable.
- **Sequential Monte Carlo** allows on-line estimation by combining Monte Carlo sampling methods with Bayesian inference.
- **Particle filter**: sequential Monte Carlo used for parameter estimation and state estimation.
  - Particle filter uses a number of independent random variables called particles, sampled directly from the state space, to represent the posterior probability
  - and update the posterior by involving the new observations;
  - the particle system is properly located, weighted, and propagated recursively according to the Bayesian rule.
- Particle filters is not the only way to tackle Bayesian filtering, e.g., differential geometry, variational methods, conjugate methods.

A few words on Particle Filters

- Kalman filtering is a special case of Bayesian filtering with linear, quadratic and Gaussian assumptions (LQG).
- We will look into the more general case of non-linear, non-Gaussian and non-stationary distributions.
- Generally for non-linear filtering no exact solution can be computed, hence we rely on numerical approximation methods.
- We will focus on sequential Monte Carlo (i.e., particle filter)

Notation

$\mathbf{y}$ — the observations

$\mathbf{x}$ — the state

N — number of samples

$\mathbf{y}_{n:0}$ — observations up to time n

$\mathbf{x}_{n:0}$ — state up to time n

$\mathbf{x}_n^{(i)}$ — i-th sample at time n

Concept of sampling

- The true distribution $P(\mathbf{x})$ can be approximated by an empirical distribution

$$\hat{P}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N}\delta(\mathbf{x}-\mathbf{x}^{(i)})$$

where $\int_X d\hat{P}(\mathbf{x}) = \int_X \hat{p}(\mathbf{x})d\mathbf{x} = 1$

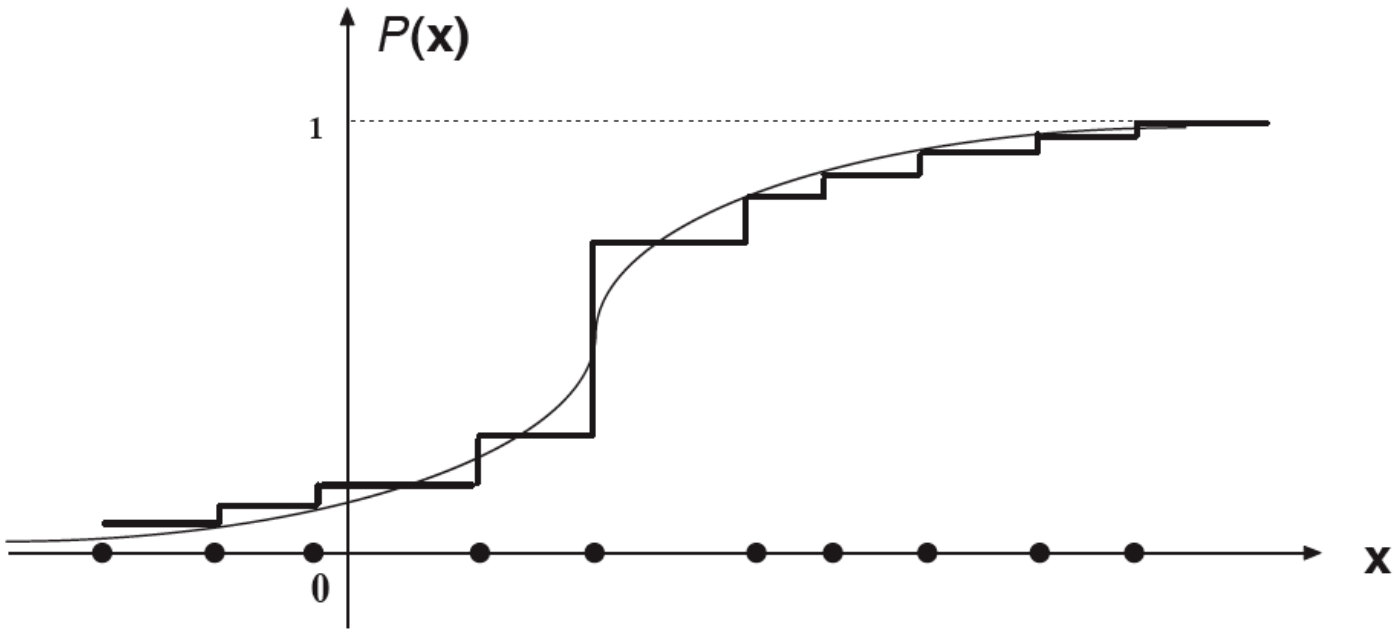

Figure: Sample approximation to the density of prob. distribution (Chen 03)

Some useful definitions

Definition

Filtering *is an operation that involves the extraction of information about a quantity of interest at time t by using data measured up to and including t.*

Definition

Prediction *derives information about what the quantity of interest will be at time $t + \tau$ in the future ($\tau > 0$) by using data measured up to and including time t.*

Some useful definitions

Definition

Filtering *is an operation that involves the extraction of information about a quantity of interest at time t by using data measured up to and including t.*

Definition

Prediction *derives information about what the quantity of interest will be at time $t + \tau$ in the future ($\tau > 0$) by using data measured up to and including time t.*

Definition

Smoothing *derives information about what the quantity of interest at time $t' < t$ by using data measured up to and including time t (i.e., in the interval $[0, t]$).*

Some useful definitions

Definition

Filtering *is an operation that involves the extraction of information about a quantity of interest at time t by using data measured up to and including t.*

Definition

Prediction *derives information about what the quantity of interest will be at time $t + \tau$ in the future ($\tau > 0$) by using data measured up to and including time t.*

Definition

Smoothing *derives information about what the quantity of interest at time $t' < t$ by using data measured up to and including time t (i.e., in the interval $[0, t]$).*

Stochastic filtering problem

- The generic stochastic filtering problem

$$\dot{\mathbf{x}}_t = \mathbf{f}(t, \mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t) \quad \text{(state equation)}$$
$$\mathbf{y}_t = \mathbf{g}(t, \mathbf{x}_t, \mathbf{u}_t, \mathbf{v}_t) \quad \text{(measurement equation)}$$

where $\mathbf{u}_t$ is the system input vector, $\mathbf{x}_t$ the state vector, $\mathbf{y}_t$ the observations, $\mathbf{w}_t$ and $\mathbf{v}_t$ are the process noise and the measurement noise, and $\mathbf{f}$ and $\mathbf{g}$ are functions which are potentially time varying.

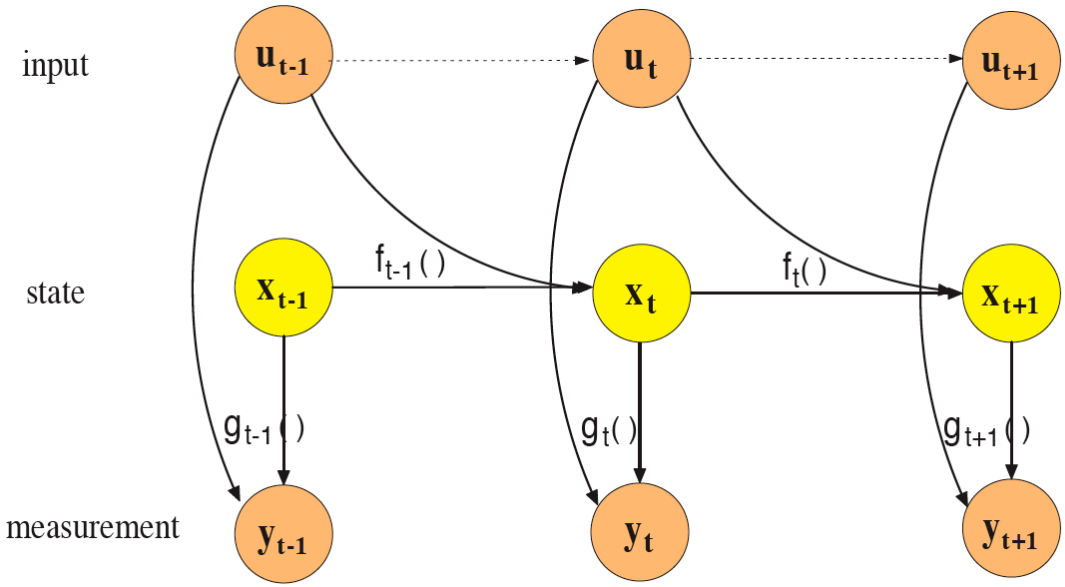

Figure: A graphical model of the state space model (Chen 03)

Simplified model: discrete case

- The generic stochastic filtering problem

$$\begin{aligned} \dot{\mathbf{x}}_t &= \mathbf{f}(t, \mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t) \quad \text{(state equation)} \\ \mathbf{y}_t &= \mathbf{g}(t, \mathbf{x}_t, \mathbf{u}_t, \mathbf{v}_t) \quad \text{(measurement equation)} \end{aligned}$$

- In practice we are interested in the discrete simplified case

$$\begin{aligned} \mathbf{x}_{n+1} &= \mathbf{f}(\mathbf{x}_n, \mathbf{w}_n) \\ \mathbf{y}_n &= \mathbf{g}(\mathbf{x}_n, \mathbf{v}_n) \end{aligned}$$

Simplified model: discrete case

- The generic stochastic filtering problem

$$\begin{aligned} \dot{\mathbf{x}}_t &= \mathbf{f}(t, \mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t) \quad \text{(state equation)} \\ \mathbf{y}_t &= \mathbf{g}(t, \mathbf{x}_t, \mathbf{u}_t, \mathbf{v}_t) \quad \text{(measurement equation)} \end{aligned}$$

- In practice we are interested in the discrete simplified case

$$\begin{aligned} \mathbf{x}_{n+1} &= \mathbf{f}(\mathbf{x}_n, \mathbf{w}_n) \\ \mathbf{y}_n &= \mathbf{g}(\mathbf{x}_n, \mathbf{v}_n) \end{aligned}$$

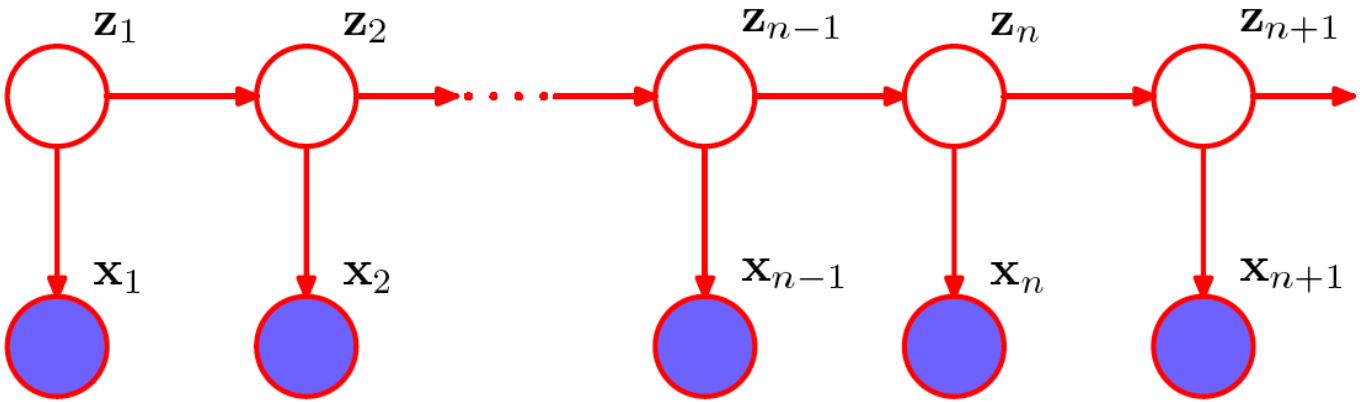

Figure: Careful today change of notation: $\mathbf{z}$ is now $\mathbf{x}$ and $\mathbf{x}$ is now $\mathbf{y}$.

Simplified model: discrete case

- The generic stochastic filtering problem

$$\begin{aligned} \dot{\mathbf{x}}_t &= \mathbf{f}(t, \mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t) \quad \text{(state equation)} \\ \mathbf{y}_t &= \mathbf{g}(t, \mathbf{x}_t, \mathbf{u}_t, \mathbf{v}_t) \quad \text{(measurement equation)} \end{aligned}$$

- In practice we are interested in the discrete simplified case

$$\begin{aligned} \mathbf{x}_{n+1} &= \mathbf{f}(\mathbf{x}_n, \mathbf{w}_n) \\ \mathbf{y}_n &= \mathbf{g}(\mathbf{x}_n, \mathbf{v}_n) \end{aligned}$$

- This equations are characterized by the state transition probability $p(\mathbf{x}_{n+1}|\mathbf{x}_n)$, and the likelihood $p(\mathbf{y}_n|\mathbf{x}_n)$.

Stochastic filtering is an inverse problem

- Given $\mathbf{y}_{n:0}$, provided $\mathbf{f}$ and $\mathbf{g}$ are known, one needs to find the best estimate $\hat{\mathbf{x}}_n$.
- This is an inverse problem: Find the inputs sequentially with a mapping function which yields the output data.
- This is an **ill-posed problem** since the inverse learning problem is one-to-many: the mapping from output to input is generally non-unique.

Definition

A problem is **well-posed** *if it satisfies: existence, uniqueness and stability.*

Intractable Bayesian problems

- **Normalization:** Given the prior $p(\mathbf{x})$ and the likelihood $p(\mathbf{y}|\mathbf{x})$, the posterior $p(\mathbf{x}|\mathbf{y})$ is obtained by dividing by the normalization factor $p(\mathbf{y})$

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{\int_X p(\mathbf{y}|\mathbf{x})p(\mathbf{x})d\mathbf{x}}$$

- **Marginalization:** Given the joint posterior, the marginal posterior

$$p(\mathbf{x}|\mathbf{y}) = \int_Z p(\mathbf{x}, \mathbf{z}|\mathbf{y})d\mathbf{z}$$

- **Expectation**

$$E_{p(\mathbf{x}|\mathbf{y})}[f(\mathbf{x})] = \int_X f(\mathbf{x})p(\mathbf{x}|\mathbf{y})d\mathbf{y}$$

Recursive Bayesian estimation I

- Let $p(\mathbf{x}_n|\mathbf{y}_{n:0})$ be the conditional pdf of $\mathbf{x}_n$

$$\begin{aligned} p(\mathbf{x}_n|\mathbf{y}_{n:0}) &= \frac{p(\mathbf{y}_{n:0}|\mathbf{x}_n)p(\mathbf{x}_n)}{p(\mathbf{y}_{n:0})} \\ &= \frac{p(\mathbf{y}_n, \mathbf{y}_{n-1:0}|\mathbf{x}_n)p(\mathbf{x}_n)}{p(\mathbf{y}_n, \mathbf{y}_{n-1:0})} \end{aligned}$$

Recursive Bayesian estimation I

- Let $p(\mathbf{x}_n|\mathbf{y}_{n:0})$ be the conditional pdf of $\mathbf{x}_n$

$$\begin{aligned} p(\mathbf{x}_n|\mathbf{y}_{n:0}) &= \frac{p(\mathbf{y}_{n:0}|\mathbf{x}_n)p(\mathbf{x}_n)}{p(\mathbf{y}_{n:0})} \\ &= \frac{p(\mathbf{y}_n, \mathbf{y}_{n-1:0}|\mathbf{x}_n)p(\mathbf{x}_n)}{p(\mathbf{y}_n, \mathbf{y}_{n-1:0})} \\ &= \frac{p(\mathbf{y}_n|\mathbf{y}_{n-1:0}, \mathbf{x}_n)p(\mathbf{y}_{n-1:0}|\mathbf{x}_n)p(\mathbf{x}_n)}{p(\mathbf{y}_n|\mathbf{y}_{n-1:0})p(\mathbf{y}_{n-1:0})} \end{aligned}$$

Recursive Bayesian estimation I

- Let $p(\mathbf{x}_n|\mathbf{y}_{n:0})$ be the conditional pdf of $\mathbf{x}_n$

$$\begin{aligned} p(\mathbf{x}_n|\mathbf{y}_{n:0}) &= \frac{p(\mathbf{y}_{n:0}|\mathbf{x}_n)p(\mathbf{x}_n)}{p(\mathbf{y}_{n:0})} \\ &= \frac{p(\mathbf{y}_n, \mathbf{y}_{n-1:0}|\mathbf{x}_n)p(\mathbf{x}_n)}{p(\mathbf{y}_n, \mathbf{y}_{n-1:0})} \\ &= \frac{p(\mathbf{y}_n|\mathbf{y}_{n-1:0}, \mathbf{x}_n)p(\mathbf{y}_{n-1:0}|\mathbf{x}_n)p(\mathbf{x}_n)}{p(\mathbf{y}_n|\mathbf{y}_{n-1:0})p(\mathbf{y}_{n-1:0})} \\ &= \frac{p(\mathbf{y}_n|\mathbf{y}_{n-1:0}, \mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{n-1:0})p(\mathbf{y}_{n-1:0})p(\mathbf{x}_n)}{p(\mathbf{y}_n|\mathbf{y}_{n-1:0})p(\mathbf{y}_{n-1:0})p(\mathbf{x}_n)} \end{aligned}$$

Recursive Bayesian estimation I

- Let $p(\mathbf{x}_n|\mathbf{y}_{n:0})$ be the conditional pdf of $\mathbf{x}_n$

$$
\begin{aligned}
p(\mathbf{x}_n|\mathbf{y}_{n:0}) &= \frac{p(\mathbf{y}_{n:0}|\mathbf{x}_n)p(\mathbf{x}_n)}{p(\mathbf{y}_{n:0})} \\
&= \frac{p(\mathbf{y}_n, \mathbf{y}_{n-1:0}|\mathbf{x}_n)p(\mathbf{x}_n)}{p(\mathbf{y}_n, \mathbf{y}_{n-1:0})} \\
&= \frac{p(\mathbf{y}_n|\mathbf{y}_{n-1:0}, \mathbf{x}_n)p(\mathbf{y}_{n-1:0}|\mathbf{x}_n)p(\mathbf{x}_n)}{p(\mathbf{y}_n|\mathbf{y}_{n-1:0})p(\mathbf{y}_{n-1:0})} \\
&= \frac{p(\mathbf{y}_n|\mathbf{y}_{n-1:0}, \mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{n-1:0})p(\mathbf{y}_{n-1:0})p(\mathbf{x}_n)}{p(\mathbf{y}_n|\mathbf{y}_{n-1:0})p(\mathbf{y}_{n-1:0})p(\mathbf{x}_n)} \\
&= \frac{p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{n-1:0})}{p(\mathbf{y}_n|\mathbf{y}_{n-1:0})}
\end{aligned}
$$

Recursive Bayesian estimation I

- Let $p(\mathbf{x}_n|\mathbf{y}_{n:0})$ be the conditional pdf of $\mathbf{x}_n$

$$
\begin{aligned}
p(\mathbf{x}_n|\mathbf{y}_{n:0}) &= \frac{p(\mathbf{y}_{n:0}|\mathbf{x}_n)p(\mathbf{x}_n)}{p(\mathbf{y}_{n:0})} \\
&= \frac{p(\mathbf{y}_n, \mathbf{y}_{n-1:0}|\mathbf{x}_n)p(\mathbf{x}_n)}{p(\mathbf{y}_n, \mathbf{y}_{n-1:0})} \\
&= \frac{p(\mathbf{y}_n|\mathbf{y}_{n-1:0}, \mathbf{x}_n)p(\mathbf{y}_{n-1:0}|\mathbf{x}_n)p(\mathbf{x}_n)}{p(\mathbf{y}_n|\mathbf{y}_{n-1:0})p(\mathbf{y}_{n-1:0})} \\
&= \frac{p(\mathbf{y}_n|\mathbf{y}_{n-1:0}, \mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{n-1:0})p(\mathbf{y}_{n-1:0})p(\mathbf{x}_n)}{p(\mathbf{y}_n|\mathbf{y}_{n-1:0})p(\mathbf{y}_{n-1:0})p(\mathbf{x}_n)} \\
&= \frac{p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{n-1:0})}{p(\mathbf{y}_n|\mathbf{y}_{n-1:0})}
\end{aligned}
$$

Recursive Bayesian estimation II

The posterior densisty is described with three terms

$$p(\mathbf{x}_n|\mathbf{y}_{n:0}) = \frac{p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{n-1:0})}{p(\mathbf{y}_n|\mathbf{y}_{n-1:0})}$$

- **Prior:** defines the knowledge of the model

$$p(\mathbf{x}_n|\mathbf{y}_{n-1:0}) = \int p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{y}_{n-1:0})d\mathbf{x}_{n-1}$$

Recursive Bayesian estimation II

The posterior densisty is described with three terms

$$p(\mathbf{x}_n|\mathbf{y}_{n:0}) = \frac{p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{n-1:0})}{p(\mathbf{y}_n|\mathbf{y}_{n-1:0})}$$

- **Prior:** defines the knowledge of the model

$$p(\mathbf{x}_n|\mathbf{y}_{n-1:0}) = \int p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{y}_{n-1:0})d\mathbf{x}_{n-1}$$

- **Likelihood:** $p(\mathbf{y}_n|\mathbf{x}_n)$ determines the measurement noise model

Recursive Bayesian estimation II

The posterior densisty is described with three terms

$$p(\mathbf{x}_n|\mathbf{y}_{n:0}) = \frac{p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{n-1:0})}{p(\mathbf{y}_n|\mathbf{y}_{n-1:0})}$$

- **Prior:** defines the knowledge of the model

$$p(\mathbf{x}_n|\mathbf{y}_{n-1:0}) = \int p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{y}_{n-1:0})d\mathbf{x}_{n-1}$$

- **Likelihood:** $p(\mathbf{y}_n|\mathbf{x}_n)$ determines the measurement noise model
- **Evidence:** which involves

$$p(\mathbf{y}_n|\mathbf{y}_{n-1:0}) = \int p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{n-1:0})d\mathbf{x}_n$$

Recursive Bayesian estimation II

The posterior densisty is described with three terms

$$p(\mathbf{x}_n|\mathbf{y}_{n:0}) = \frac{p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{n-1:0})}{p(\mathbf{y}_n|\mathbf{y}_{n-1:0})}$$

- **Prior:** defines the knowledge of the model

$$p(\mathbf{x}_n|\mathbf{y}_{n-1:0}) = \int p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{y}_{n-1:0})d\mathbf{x}_{n-1}$$

- **Likelihood:** $p(\mathbf{y}_n|\mathbf{x}_n)$ determines the measurement noise model
- **Evidence:** which involves

$$p(\mathbf{y}_n|\mathbf{y}_{n-1:0}) = \int p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{n-1:0})d\mathbf{x}_n$$

We need to define a criteria for optimal filtering

Recursive Bayesian estimation II

The posterior densisty is described with three terms

$$p(\mathbf{x}_n|\mathbf{y}_{n:0}) = \frac{p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{n-1:0})}{p(\mathbf{y}_n|\mathbf{y}_{n-1:0})}$$

- **Prior:** defines the knowledge of the model

$$p(\mathbf{x}_n|\mathbf{y}_{n-1:0}) = \int p(\mathbf{x}_n|\mathbf{x}_{n-1})p(\mathbf{x}_{n-1}|\mathbf{y}_{n-1:0})d\mathbf{x}_{n-1}$$

- **Likelihood:** $p(\mathbf{y}_n|\mathbf{x}_n)$ determines the measurement noise model
- **Evidence:** which involves

$$p(\mathbf{y}_n|\mathbf{y}_{n-1:0}) = \int p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{n-1:0})d\mathbf{x}_n$$

We need to define a criteria for optimal filtering

Criteria for optimal filtering I

An optimal filter is "optimal" under a particular criteria

- **Minimum mean-squared error (MMSE):** defined in terms of prediction or filtering error

$$E[||\mathbf{x}_n - \hat{\mathbf{x}}_n||_2^2 | \mathbf{y}_{n:0}] = \int ||\mathbf{x}_n - \hat{\mathbf{x}}_n||_2^2 p(\mathbf{x}_n|\mathbf{y}_{n:0}) d\mathbf{x}_n$$

  which is aimed to find the *conditional mean*

$$\hat{\mathbf{x}}_n = E[\mathbf{x}_n|\mathbf{y}_{n:0}] = \int \mathbf{x}_n p(\mathbf{x}_n|\mathbf{y}_{n:0}) d\mathbf{x}_n$$

- **Maximum a posteriori (MAP):** It is aimed to find the mode of posterior probability $p(\mathbf{x}_n|\mathbf{y}_{n:0})$

Criteria for optimal filtering I

An optimal filter is "optimal" under a particular criteria

- **Minimum mean-squared error (MMSE):** defined in terms of prediction or filtering error

$$E[||\mathbf{x}_n - \hat{\mathbf{x}}_n||_2^2|\mathbf{y}_{n:0}] = \int ||\mathbf{x}_n - \hat{\mathbf{x}}_n||_2^2 p(\mathbf{x}_n|\mathbf{y}_{n:0}) d\mathbf{x}_n$$

  which is aimed to find the *conditional mean*

$$\hat{\mathbf{x}}_n = E[\mathbf{x}_n|\mathbf{y}_{n:0}] = \int \mathbf{x}_n p(\mathbf{x}_n|\mathbf{y}_{n:0}) d\mathbf{x}_n$$

- **Maximum a posteriori (MAP):** It is aimed to find the mode of posterior probability $p(\mathbf{x}_n|\mathbf{y}_{n:0})$
- **Maximum likelihood (ML):** which reduces to a special case of MAP where the prior is neglected.

Criteria for optimal filtering I

An optimal filter is "optimal" under a particular criteria

- **Minimum mean-squared error (MMSE):** defined in terms of prediction or filtering error

$$E[||\mathbf{x}_n - \hat{\mathbf{x}}_n||_2^2|\mathbf{y}_{n:0}] = \int ||\mathbf{x}_n - \hat{\mathbf{x}}_n||_2^2 p(\mathbf{x}_n|\mathbf{y}_{n:0}) d\mathbf{x}_n$$

  which is aimed to find the *conditional mean*

$$\hat{\mathbf{x}}_n = E[\mathbf{x}_n|\mathbf{y}_{n:0}] = \int \mathbf{x}_n p(\mathbf{x}_n|\mathbf{y}_{n:0}) d\mathbf{x}_n$$

- **Maximum a posteriori (MAP):** It is aimed to find the mode of posterior probability $p(\mathbf{x}_n|\mathbf{y}_{n:0})$
- **Maximum likelihood (ML):** which reduces to a special case of MAP where the prior is neglected.
- **Minimax:** which is to find the median of posterior $p(\mathbf{x}_n|\mathbf{y}_{n:0})$.

Criteria for optimal filtering I

An optimal filter is "optimal" under a particular criteria

- **Minimum mean-squared error (MMSE):** defined in terms of prediction or filtering error

$$E[||\mathbf{x}_n - \hat{\mathbf{x}}_n||_2^2|\mathbf{y}_{n:0}] = \int ||\mathbf{x}_n - \hat{\mathbf{x}}_n||_2^2 p(\mathbf{x}_n|\mathbf{y}_{n:0}) d\mathbf{x}_n$$

  which is aimed to find the *conditional mean*

$$\hat{\mathbf{x}}_n = E[\mathbf{x}_n|\mathbf{y}_{n:0}] = \int \mathbf{x}_n p(\mathbf{x}_n|\mathbf{y}_{n:0}) d\mathbf{x}_n$$

- **Maximum a posteriori (MAP):** It is aimed to find the mode of posterior probability $p(\mathbf{x}_n|\mathbf{y}_{n:0})$
- **Maximum likelihood (ML):** which reduces to a special case of MAP where the prior is neglected.
- **Minimax:** which is to find the median of posterior $p(\mathbf{x}_n|\mathbf{y}_{n:0})$.

Criteria for optimal filtering II

- MMSE: finds the mean
- MAP: finds the mode
- Minimax: finds the median

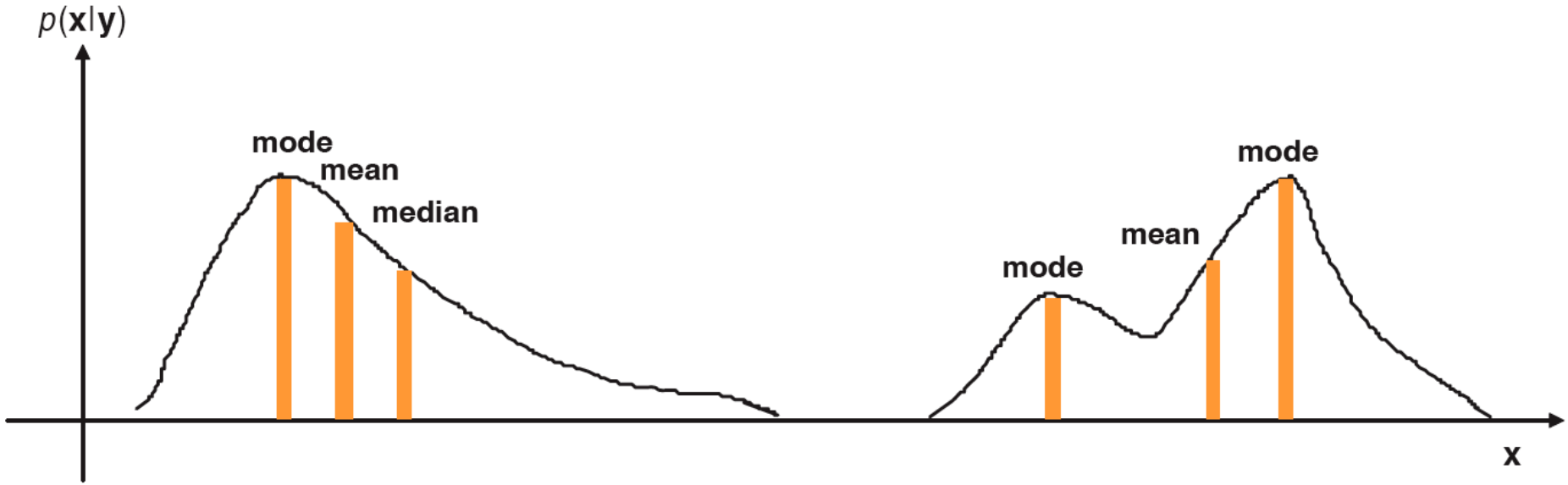

Figure: (left) Three optimal criteria that seek different solutions for a skewed unimodal distribution (right) MAP is misleading for the multimodal distribution (Chen 03)

Criteria for optimal filtering III

An optimal filter is "optimal" under a particular criteria

- **Minimum conditional inaccuracy:** defined as

$$E_{p(\mathbf{x},\mathbf{y})}[-\log \hat{p}(\mathbf{x}|\mathbf{y})] = \int p(\mathbf{x},\mathbf{y}) \log \frac{1}{\hat{p}(\mathbf{x}|\mathbf{y})} d\mathbf{x}d\mathbf{y}$$

- **Minimum conditional KL divergence**

$$KL(p||\hat{p}) = \int p(\mathbf{x},\mathbf{y}) \log \frac{p(\mathbf{x},\mathbf{y})}{\hat{p}(\mathbf{x}|\mathbf{y})p(\mathbf{x})} d\mathbf{x}d\mathbf{y}$$

where the KL is a measure of divergence between distributions such that $0 \leq KL(p||\hat{p}) \leq 1$. The KL is 0 only when the distributions are the same.

Criteria for optimal filtering III

An optimal filter is "optimal" under a particular criteria

- **Minimum conditional inaccuracy:** defined as

$$E_{p(\mathbf{x},\mathbf{y})}[-\log \hat{p}(\mathbf{x}|\mathbf{y})] = \int p(\mathbf{x},\mathbf{y}) \log \frac{1}{\hat{p}(\mathbf{x}|\mathbf{y})} d\mathbf{x}d\mathbf{y}$$

- **Minimum conditional KL divergence**

$$KL(p||\hat{p}) = \int p(\mathbf{x},\mathbf{y}) \log \frac{p(\mathbf{x},\mathbf{y})}{\hat{p}(\mathbf{x}|\mathbf{y})p(\mathbf{x})} d\mathbf{x}d\mathbf{y}$$

where the KL is a measure of divergence between distributions such that $0 \leq KL(p||\hat{p}) \leq 1$. The KL is 0 only when the distributions are the same.

Criteria for optimal filtering IV

An optimal filter is "optimal" under a particular criteria

- **Minimum free energy:** It is a lower bound of maximum log-likelihood, which is aimed to minimize

$$\begin{aligned}\mathcal{F}(Q;P) &\equiv E_{Q(\mathbf{x})}[-\log P(\mathbf{x}|\mathbf{y})] \\ &= E_{Q(\mathbf{x})}[\log \frac{Q(\mathbf{x})}{P(\mathbf{x}|\mathbf{y})}] - E_{Q(\mathbf{x})}[\log Q(\mathbf{x})] \\ &= KL(Q||P) - H(Q)\end{aligned}$$

This minimization can be done using (EM) algorithm

$$\begin{aligned}Q(\mathbf{x}_{n+1}) &\leftarrow \underset{Q}{\operatorname{argmax}}\, \mathcal{F}(Q;P) \\ \mathbf{x}_{n+1} &\leftarrow \underset{\mathbf{x}}{\operatorname{argmax}}\, \mathcal{F}(Q;P)\end{aligned}$$

Which criteria to choose?

- All these criteria are valid for state and parameter estimation
- MMSE requires the computation of the prior, likelihood and evidence.
- MAP requires the computation of the prior and likelihood, but not the denominator (integration) and thereby more computational inexpensive;
- MAP estimate has a drawback especially in a high-dimensional space. High probability density does not imply high probability mass.
- A narrow spike with very small width (support) can have a very high density, but the actual probability of estimated state belonging to it is small.
- Hence, the width of the mode is more important than its height in the high-dimensional case.
- The last three criteria are all ML oriented. They are very related.

Bayesian filtering

- The criterion of optimality used for **Bayesian filtering** is the Bayes risk of MMSE

$$E[||\mathbf{x}_n - \hat{\mathbf{x}}_n||_2^2 | \mathbf{y}_{n:0}] = \int ||\mathbf{x}_n - \hat{\mathbf{x}}_n||_2^2 p(\mathbf{x}_n|\mathbf{y}_{n:0}) d\mathbf{x}_n$$

- Bayesian filtering is optimal in a sense that it seeks the posterior distribution which integrates and uses all of available information expressed by probabilities
- As time proceeds, one needs infinite computing power and unlimited memory to calculate the optimal solution, except in some special cases (e.g. linear Gaussian).
- In general we can only seek a suboptimal or locally optimal solution.

Kalman filter revisited

- In practice we are interested in the discrete simplified case

$$\begin{aligned} \mathbf{x}_{n+1} &= \mathbf{f}(\mathbf{x}_n, \mathbf{w}_n) \\ \mathbf{y}_n &= \mathbf{g}(\mathbf{x}_n, \mathbf{v}_n) \end{aligned}$$

- When the dynamic system is linear Gaussian this reduces to

$$\begin{aligned} \mathbf{x}_{n+1} &= \mathbf{F}_{n+1,n}\mathbf{x}_n + \mathbf{w}_n \\ \mathbf{y}_n &= \mathbf{G}_n\mathbf{x}_n + \mathbf{v}_n \end{aligned}$$

  with $\mathbf{F}_{n+1,n}$ the transition matrix, and $\mathbf{G}_n$ the measurement matrix.

- This is the **Kalman filter**, and we saw that by propagating sufficient statistics (i.e., mean and covariance) we can solve the system analytically.
- In the general case it is not tractable, and we will rely on approximations.

Kalman filter: Forward equations I

- We start by defining the messages

$$\hat{\alpha}(\mathbf{z}_n) = \mathcal{N}(\mathbf{z}_n|\boldsymbol{\mu}_n, \mathbf{V}_n)$$

- Using the HMM recursion formulas for continuous variables we have

$$c_n\hat{\alpha}(\mathbf{z}_n) = p(\mathbf{x}_n|\mathbf{z}_n)\int \hat{\alpha}(\mathbf{z}_{n-1})p(\mathbf{z}_n|\mathbf{z}_{n-1})d\mathbf{z}_{n-1}$$

- Substituting the conditionals we have

$$\begin{aligned} c_n\mathcal{N}(\mathbf{z}_n|\boldsymbol{\mu}_n, \mathbf{V}_n) &= \mathcal{N}(\mathbf{x}_n|\mathbf{C}\mathbf{z}_n, \boldsymbol{\Sigma})\int \mathcal{N}(\mathbf{z}_{n-1}|\boldsymbol{\mu}_{n-1}, \mathbf{V}_{n-1})\mathcal{N}(\mathbf{z}_n|\mathbf{A}\mathbf{x}_{n-1}, \boldsymbol{\Gamma})d\mathbf{z}_{n-1} \\ &= \mathcal{N}(\mathbf{x}_n|\mathbf{C}\mathbf{z}_n, \boldsymbol{\Sigma})\mathcal{N}(\mathbf{z}_n|\mathbf{A}\boldsymbol{\mu}_{n-1}, \mathbf{P}_{n-1}) \end{aligned}$$

- Here we assume that $\boldsymbol{\mu}_{n-1}$, and $\mathbf{V}_{n-1}$ are known, and we have defined

$$\mathbf{P}_{n-1} = \mathbf{A}\mathbf{V}_{n-1}\mathbf{A}^T + \boldsymbol{\Gamma}$$

Kalman filter: Forward equations II

- Given the values of $\boldsymbol{\mu}_{n-1}$, $\mathbf{V}_{n-1}$ and the new observation $\mathbf{x}_n$, we can evaluate the Gaussian marginal for $\mathbf{z}_n$ having mean $\boldsymbol{\mu}_n$ and covariance $\mathbf{V}_n$ as well as the normalization coefficient c_n

$$\begin{aligned} \boldsymbol{\mu}_n &= \mathbf{A}\boldsymbol{\mu}_{n-1} + \mathbf{K}_n(\mathbf{x}_n - \mathbf{C}\mathbf{A}\boldsymbol{\mu}_{n-1}) \\ \mathbf{V}_n &= (\mathbf{I} - \mathbf{K}_n\mathbf{C})\mathbf{P}_{n-1} \\ c_n &= \mathcal{N}(\mathbf{x}_n|\mathbf{C}\mathbf{A}\boldsymbol{\mu}_{n-1}, \mathbf{C}\mathbf{P}_{n-1}\mathbf{C}^T + \boldsymbol{\Sigma}) \end{aligned}$$

  where the **Kalman gain matrix** is defined as

$$\mathbf{K}_n = \mathbf{P}_{n-1}\mathbf{C}^T(\mathbf{C}\mathbf{P}_{n-1}\mathbf{C}^T + \boldsymbol{\Sigma})^{-1}$$

- The initial conditions are given by

$$\boldsymbol{\mu}_1 = \boldsymbol{\mu}_0 + \mathbf{K}_1(\mathbf{x}_1 - \mathbf{C}\boldsymbol{\mu}_0) \qquad \mathbf{V}_1 = (\mathbf{I} - \mathbf{K}_1\mathbf{C})\mathbf{V}_0$$
$$c_1 = \mathcal{N}(\mathbf{x}_1|\mathbf{C}\boldsymbol{\mu}_0, \mathbf{C}\mathbf{V}_0\mathbf{C}^T + \boldsymbol{\Sigma}) \qquad \mathbf{K}_1 = \mathbf{V}_0\mathbf{C}^T(\mathbf{C}\mathbf{V}_0\mathbf{C}^T + \boldsymbol{\Sigma})^{-1}$$

- Interpretation is making prediction and doing corrections with $\mathbf{K}_n$.
- The likelihood can be computed as $p(\mathbf{X}) = \prod_{n=1}^{N} c_n$.

Optimum non-linear filters

- The use of Kalman filtering is limited by the ubiquitous nonlinearity and non-Gaussianity of physical world.
- The nonlinear filtering consists in finding $p(\mathbf{x}|\mathbf{y}_{n:0})$.
- The number of variables is infinite, but not all of them are of equal importance.
- **Global approach:** one attempts to solve a PDE instead of an ODE in linear case. Numerical approximation techniques are needed to solve the equation.
- **Local approach:** finite sum approximation (e.g. Gaussian sum filter), linearization techniques (i.e. EKF) or numerical approximations (e.g., particle filter) are usually used.

Extended Kalman filter (EKF)

- Recall the equations of motion

$$\begin{aligned} \mathbf{x}_{n+1} &= \mathbf{f}(\mathbf{x}_n, \mathbf{w}_n) \\ \mathbf{y}_n &= \mathbf{g}(\mathbf{x}_n, \mathbf{v}_n) \end{aligned}$$

- These equations are linearized in the EKF

$$\hat{\mathbf{F}}_{n+1,n} = \left.\frac{d\mathbf{f}(\mathbf{x})}{d\mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_n}, \qquad \hat{\mathbf{G}}_{n+1,n} = \left.\frac{d\mathbf{g}(\mathbf{x})}{d\mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{n|n-1}}$$

- Then the conventional Kalman filter can be employed.
- Because EKF always approximates the posterior $p(\mathbf{x}_n|\mathbf{y}_{n:0})$ as a Gaussian, provides poor performance when the true posterior is non-Gaussian (e.g. heavily skewed or multimodal).
- A more general solution is to rely on numerical approximations.

Numerical approximations

- Monte-carlo sampling approximation (i.e., particle filter)
- Gaussian/Laplace approximation
- Iterative quadrature
- Multi-grid method and point-mass approximation
- Moment approximation
- Gaussian sum approximation
- Deterministic sampling approximation

Monte Carlo sampling

- It's brute force technique that provided that one can drawn i.i.d. samples $\{\mathbf{x}^{(1)} \cdots \mathbf{x}^{N}\}$ from probability distribution $P(\mathbf{x})$ so that

$$\int_X f(\mathbf{x}) dP(\mathbf{x}) \approx \frac{1}{N} \sum_{i=1}^{N} f\left(\mathbf{x}^{(i)}\right) = \hat{f}_N$$

  for which $E[\hat{f}_N] = E[f]$ and $\text{Var}[\hat{f}_N] = \frac{1}{N}\text{Var}[f] = \frac{\sigma^2}{N}$

- By the *Kolmogorov Strong Law of Large Numbers* (under some mild regularity conditions), $\hat{f}_N(\mathbf{x})$ converges to $E[f(\mathbf{x})]$ with high probability.

Monte Carlo sampling

- It's brute force technique that provided that one can drawn i.i.d. samples $\{\mathbf{x}^{(1)} \cdots \mathbf{x}^{N}\}$ from probability distribution $P(\mathbf{x})$ so that

$$\int_X f(\mathbf{x}) dP(\mathbf{x}) \approx \frac{1}{N} \sum_{i=1}^{N} f\left(\mathbf{x}^{(i)}\right) = \hat{f}_N$$

  for which $E[\hat{f}_N] = E[f]$ and $\text{Var}[\hat{f}_N] = \frac{1}{N}\text{Var}[f] = \frac{\sigma^2}{N}$

- By the *Kolmogorov Strong Law of Large Numbers* (under some mild regularity conditions), $\hat{f}_N(\mathbf{x})$ converges to $E[f(\mathbf{x})]$ with high probability.
- The convergence rate is assessed by the *Central Limit Theorem*

$$\sqrt{N}\left(\hat{f}_N - E[f]\right) \sim \mathcal{N}(0, \sigma^2)$$

  where σ^2 is the variance of $f(\mathbf{x})$. The error rate is of order $\mathcal{O}(N^{-1/2})$.

Monte Carlo sampling

- It's brute force technique that provided that one can drawn i.i.d. samples $\{\mathbf{x}^{(1)} \cdots \mathbf{x}^{N}\}$ from probability distribution $P(\mathbf{x})$ so that

$$\int_X f(\mathbf{x}) dP(\mathbf{x}) \approx \frac{1}{N} \sum_{i=1}^{N} f\left(\mathbf{x}^{(i)}\right) = \hat{f}_N$$

  for which $E[\hat{f}_N] = E[f]$ and $\text{Var}[\hat{f}_N] = \frac{1}{N}\text{Var}[f] = \frac{\sigma^2}{N}$

- By the *Kolmogorov Strong Law of Large Numbers* (under some mild regularity conditions), $\hat{f}_N(\mathbf{x})$ converges to $E[f(\mathbf{x})]$ with high probability.
- The convergence rate is assessed by the *Central Limit Theorem*

$$\sqrt{N}\left(\hat{f}_N - E[f]\right) \sim \mathcal{N}(0, \sigma^2)$$

  where σ^2 is the variance of $f(\mathbf{x})$. The error rate is of order $\mathcal{O}(N^{-1/2})$.
- An important property is that the estimation accuracy is independent of the dimensionality of the state space.
- The variance of estimate is inversely proportional to the number of samples.

Monte Carlo sampling

- It's brute force technique that provided that one can drawn i.i.d. samples $\{\mathbf{x}^{(1)} \cdots \mathbf{x}^{N}\}$ from probability distribution $P(\mathbf{x})$ so that

$$\int_X f(\mathbf{x}) dP(\mathbf{x}) \approx \frac{1}{N} \sum_{i=1}^{N} f\left(\mathbf{x}^{(i)}\right) = \hat{f}_N$$

  for which $E[\hat{f}_N] = E[f]$ and $\text{Var}[\hat{f}_N] = \frac{1}{N}\text{Var}[f] = \frac{\sigma^2}{N}$

- By the *Kolmogorov Strong Law of Large Numbers* (under some mild regularity conditions), $\hat{f}_N(\mathbf{x})$ converges to $E[f(\mathbf{x})]$ with high probability.
- The convergence rate is assessed by the *Central Limit Theorem*

$$\sqrt{N}\left(\hat{f}_N - E[f]\right) \sim \mathcal{N}(0, \sigma^2)$$

  where σ^2 is the variance of $f(\mathbf{x})$. The error rate is of order $\mathcal{O}(N^{-1/2})$.
- An important property is that the estimation accuracy is independent of the dimensionality of the state space.
- The variance of estimate is inversely proportional to the number of samples.

Fundamental problems of Monte Carlo estimation

Monte carlo methods approximate

$$\int_X f(\mathbf{x}) dP(\mathbf{x}) \approx \frac{1}{N} \sum_{i=1}^{N} f\left(\mathbf{x}^{(i)}\right) = \hat{f}_N$$

There are two fundamental problems:

- How to drawn samples from a probability distribution $P(\mathbf{x})$?
- How to estimate the expectation of a function w.r.t. the distribution or density, i.e., $E[f(\mathbf{x})] = \int f(\mathbf{x}) dP(\mathbf{x})$?

Important properties of an estimator

- **Consistency:** An estimator is consistent if the estimator converges to the true value with high probability as the number of observations approaches infinity
- **Unbiasedness:** An estimator is unbiased if its expected value is equal to the true value.
- **Efficiency:** An estimator is efficient if it produces the smallest error covariance matrix among all unbiased estimators.
- **Robustness:** An estimator is robust if it is insensitive to the gross measurement errors and the uncertainties of the model.
- **Minimal variance**

Types of Monte Carlo sampling

- Importance sampling (IS)
- Rejection sampling
- Sequential importance sampling
- Sampling-importance resampling
- Stratified sampling
- Markov chain Monte Carlo (MCMC): Metropolis-Hastings and Gibbs sampling
- Hybrid Monte Carlo (HMC)
- Quasi-Monte Carlo (QMC)

Importance Sampling I

- Sample the distribution in the region of importance in order to achieve computational efficiency.
- This is important for the high-dimensional space where the data is sparse, and the region of interest where the target lies in is relatively small.
- The idea is to choose a proposal distribution $q(\mathbf{x})$ in place of the true probability distribution $p(\mathbf{x})$, which is hard-to-sample.

$$\int f(\mathbf{x})p(\mathbf{x})d\mathbf{x} = \int f(\mathbf{x})\frac{p(\mathbf{x})}{q(\mathbf{x})}q(\mathbf{x})d\mathbf{x}$$

Importance Sampling I

- Sample the distribution in the region of importance in order to achieve computational efficiency.
- This is important for the high-dimensional space where the data is sparse, and the region of interest where the target lies in is relatively small.
- The idea is to choose a proposal distribution $q(\mathbf{x})$ in place of the true probability distribution $p(\mathbf{x})$, which is hard-to-sample.

$$\int f(\mathbf{x})p(\mathbf{x})d\mathbf{x} = \int f(\mathbf{x})\frac{p(\mathbf{x})}{q(\mathbf{x})}q(\mathbf{x})d\mathbf{x}$$

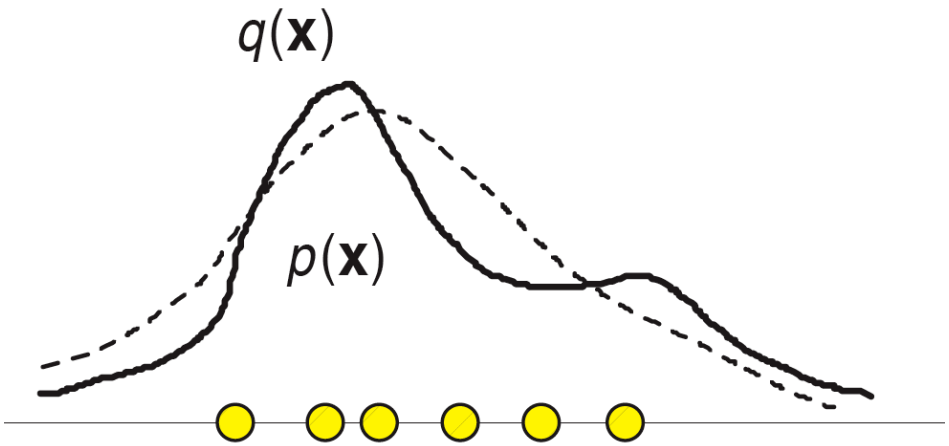

Figure: Importance sampling (Chen 03)

Importance Sampling I

- Sample the distribution in the region of importance in order to achieve computational efficiency.
- This is important for the high-dimensional space where the data is sparse, and the region of interest where the target lies in is relatively small.
- The idea is to choose a proposal distribution $q(\mathbf{x})$ in place of the true probability distribution $p(\mathbf{x})$, which is hard-to-sample.

$$\int f(\mathbf{x})p(\mathbf{x})d\mathbf{x} = \int f(\mathbf{x})\frac{p(\mathbf{x})}{q(\mathbf{x})}q(\mathbf{x})d\mathbf{x}$$

- Monte Carlo importance sampling uses N independent samples drawn from $q(\mathbf{x})$ to approximate

$$\hat{f} = \frac{1}{N}\sum_{i=1}^{N} W(\mathbf{x}^{(i)})f(\mathbf{x}^{(i)})$$

where $W(\mathbf{x}^{(i)}) = p(\mathbf{x}^{(i)})/q(\mathbf{x}^{(i)})$ are called the *importance weights*.

Importance Sampling II

- If the normalizing factor of $p(\mathbf{x})$ is not known, the importance weights can be only evaluated up to a normalizing constant.
- To ensure that we importance weights are normalized

$$\hat{f} = \sum_{i=1}^{N} \tilde{W}(\mathbf{x}^{(i)}) f(\mathbf{x}^{(i)}) \qquad \text{with} \qquad \tilde{W}(\mathbf{x}^{(i)}) = \frac{W(\mathbf{x}^{(i)})}{\sum_{i=1}^{N} W(\mathbf{x}^{(i)})}$$

- The variance of the estimate is given by

$$\begin{aligned} \text{Var}[\hat{f}] &= \frac{1}{N}\text{Var}[f(\mathbf{x})W(\mathbf{x})] = \frac{1}{N}\text{Var}[f(\mathbf{x})\frac{p(\mathbf{x})}{q(\mathbf{x})}] \\ &= \frac{1}{N}\int \left(\frac{f(\mathbf{x})p(\mathbf{x})}{q(\mathbf{x})}\right)^2 d\mathbf{x} - \frac{(E[f(\mathbf{x})])^2}{N} \end{aligned}$$

- The variance can be reduced when $q(\mathbf{x})$ is chosen to
  - match the shape of $p(\mathbf{x})$ so as to approximate the true variance
  - match the shape of $|f(\mathbf{x})|p(\mathbf{x})$ so as to further reduce the true variance
- The estimator is *biased* but *consistent*

Remarks on importance sampling

- It provides an elegant way to reduce the variance of the estimator (possibly even less than the true variance)
- it can be used when encountering the difficulty to sample from the true probability distribution directly.
- The proposal distribution $q(\mathbf{x})$ should have a heavy tail so as to be insensitive to the outliers.
- If $q(\cdot)$ is not close to $p(\cdot)$, the weights are very uneven, thus many samples are almost useless because of their negligible contributions.
- In a high-dimensional space, the importance sampling estimate is likely dominated by a few samples with large importance weights.
- Importance sampler can be mixed with Gibbs sampling or Metropolis-Hastings algorithm to produce more efficient techniques

Rejection sampling

- Rejection sampling is useful when we know (pointwise) the upper bound of underlying distribution or density.
- Assume there exists a known constant $C < \infty$ such that $p(\mathbf{x}) < Cq(\mathbf{x})$ for every $\mathbf{x} \in X$, the sampling

for $n = 1$ to N **do**
  Sample $u \sim \mathcal{U}(0, 1)$
  Sample $\mathbf{x} \sim q(\mathbf{x})$
  if $u > \frac{p(\mathbf{x})}{Cq(\mathbf{x})}$ **then**
    Repeat sampling
  end if
end for

Rejection sampling

- Rejection sampling is useful when we know (pointwise) the upper bound of underlying distribution or density.
- Assume there exists a known constant $C < \infty$ such that $p(\mathbf{x}) < Cq(\mathbf{x})$ for every $\mathbf{x} \in X$, the sampling

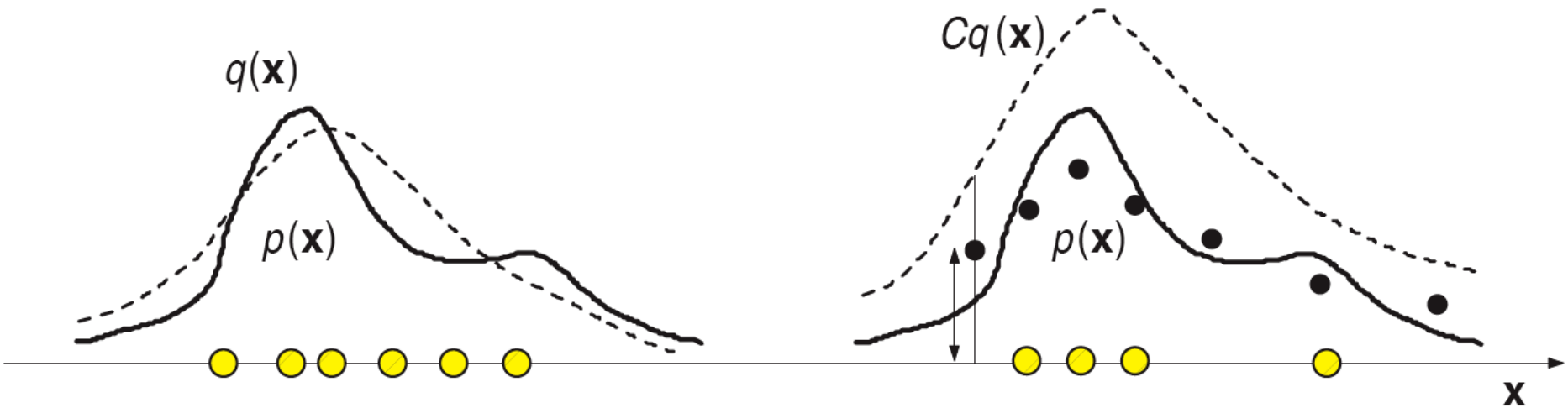

Figure: Importance (left) and Rejection (right) sampling (Chen 03)

Rejection sampling

- Rejection sampling is useful when we know (pointwise) the upper bound of underlying distribution or density.
- Assume there exists a known constant $C < \infty$ such that $p(\mathbf{x}) < Cq(\mathbf{x})$ for every $\mathbf{x} \in X$, the sampling
- The acceptance probability for a random variable is inversely proportional to the constant C.
- The choice of C is critical:
  - if $C \ll$ the samples are not reliable because of low rejection rate
  - if $C \gg$ inefficient sampling since the acceptance rate will be low
- If the prior $p(\mathbf{x})$ is used as $q(\mathbf{x})$, and the likelihood $p(\mathbf{y}|\mathbf{x}) \leq C$ and C is known, then

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y})} \leq \frac{Cq(\mathbf{x})}{p(\mathbf{y})} \equiv C'q(\mathbf{x})$$

and the acceptance rate for sample $\mathbf{x}$ is $\frac{p(\mathbf{x}|\mathbf{y})}{C'q(\mathbf{x})} = \frac{p(\mathbf{y}|\mathbf{x})}{C}$

Remarks on rejection sampling

- The draws obtained from rejection sampling are exact.
- The prerequisite of rejection sampling is the prior knowledge of constant C, which is sometimes unavailable.
- It usually takes a long time to get the samples when the ratio $p(\mathbf{x})/Cq(\mathbf{x})$ is close to zero

Sequential Importance Sampling I

- A good proposal distribution is essential to the efficiency of importance sampling...
- ... but it is usually difficult to find a good proposal distribution especially in a high-dimensional space.
- A natural way to alleviate this problem is to construct the proposal distribution sequentially, this is **sequential importance sampling**.
- if the proposal distribution is chosen in a factorized form

$$q(\mathbf{x}_{n:0}|\mathbf{y}_{n:0}) = q(\mathbf{x}_0)\prod_{t=1}^{n} q(\mathbf{x}_t|\mathbf{x}_{t-1:0}, \mathbf{y}_{t:0})$$

then the importance sampling can be performed recursively.

Sequential Importance Sampling II

- According to the telescope law of probability, we have

$$\begin{aligned} p(\mathbf{x}_{n:0}) &= p(\mathbf{x}_0)p(\mathbf{x}_1|\mathbf{x}_0)\cdots p(\mathbf{x}_n|\mathbf{x}_0,\cdots,\mathbf{x}_{n-1}) \\ q(\mathbf{x}_{n:0}) &= q_0(\mathbf{x}_0)q_1(\mathbf{x}_1|\mathbf{x}_0)\cdots q_n(\mathbf{x}_n|\mathbf{x}_0,\cdots,\mathbf{x}_{n-1}) \end{aligned}$$

- The weights can be recursively calculated as

$$W_n(\mathbf{x}_{n:0}) = \frac{p(\mathbf{x}_{n:0})}{q(\mathbf{x}_{n:0})} = W_{n-1}(\mathbf{x}_{n:0})\frac{p(\mathbf{x}_n|\mathbf{x}_{n-1:0})}{q_n(\mathbf{x}_n|\mathbf{x}_{n-1:0})}$$

Remarks on Sequential Importance Sampling

- The advantage of SIS is that it doesnt rely on the underlying Markov chain.
- Many i.i.d. replicates are run to create an importance sampler, which consequently improves the efficiency.
- The disadvantage of SIS is that the importance weights may have large variances, resulting in inaccurate estimate.
- The variance of the importance weights increases over time, **weight degeneracy problem**, after a few iterations of algorithm, only few or one of $W(\mathbf{x}^{(i)})$ will be nonzero.
- We will see now that in order to cope with this situation, resampling step is suggested to be used after weight normalization.

Sampling Importance Resampling (SIR)

- The idea is to evaluate the properties of an estimator through the empirical cumulative distribution function (cdf) of the samples instead of the true cdf.
- The resampling step is aimed to eliminate the samples with small importance weights and duplicate the samples with big weights.

Sample N random samples $\{\mathbf{x}^{(i)}\}_{i=1}^{N}$ from $q(\mathbf{x})$
for $i = 1, \cdots, N$ **do**
$W^{(i)} \propto \frac{p(\mathbf{x}^{(i)})}{q(\mathbf{x}^{(i)})}$
end for
for $i = 1, \cdots, N$ **do**
Normalize weights $\tilde{W}(\mathbf{x}^{(i)}) = \frac{W(\mathbf{x}^{(i)})}{\sum_{i=1}^{N} W(\mathbf{x}^{(i)})}$
end for
Resample with replacement N times from the discrete set $\{\mathbf{x}^{(i)}\}_{i=1}^{N}$, where the probability of resampling from each $\mathbf{x}^{(i)}$ is proportional to $\tilde{W}(\mathbf{x}^{(i)})$.

Remarks on Sampling Importance Resampling

- Resampling can be taken at every step or only taken if regarded necessary.
  - **Deterministic:** resampling is taken at every k time step (usually $k = 1$).
  - **Dynamic:** resampling is taken only when the variance of the importance weights is over the threshold.
- The particles and associated importance weights $\{\mathbf{x}^{(i)}, \tilde{W}^{(i)}\}$ are replaced by the new samples with equal importance weights (i.e. $\tilde{W}^{(i)} = 1/N$).

Remarks on Sampling Importance Resampling

- Resampling can be taken at every step or only taken if regarded necessary.
  - **Deterministic:** resampling is taken at every k time step (usually $k = 1$).
  - **Dynamic:** resampling is taken only when the variance of the importance weights is over the threshold.
- The particles and associated importance weights $\{\mathbf{x}^{(i)}, \tilde{W}^{(i)}\}$ are replaced by the new samples with equal importance weights (i.e. $\tilde{W}^{(i)} = 1/N$).
- Resampling is important because
  - if importance weights are uneven distributed, propagating the trivial weights through the dynamic system is a waste of computing power;
  - when the importance weights are skewed, resampling can provide chances for selecting important samples and rejuvenate the sampler

Remarks on Sampling Importance Resampling

- Resampling can be taken at every step or only taken if regarded necessary.
  - **Deterministic:** resampling is taken at every k time step (usually $k = 1$).
  - **Dynamic:** resampling is taken only when the variance of the importance weights is over the threshold.
- The particles and associated importance weights $\{\mathbf{x}^{(i)}, \tilde{W}^{(i)}\}$ are replaced by the new samples with equal importance weights (i.e. $\tilde{W}^{(i)} = 1/N$).
- Resampling is important because
  - if importance weights are uneven distributed, propagating the trivial weights through the dynamic system is a waste of computing power;
  - when the importance weights are skewed, resampling can provide chances for selecting important samples and rejuvenate the sampler
- Resampling does not necessarily improve the current state estimate because it also introduces extra Monte Carlo variation.
- There are many types of resampling methods.

Remarks on Sampling Importance Resampling

- Resampling can be taken at every step or only taken if regarded necessary.
  - **Deterministic:** resampling is taken at every k time step (usually $k = 1$).
  - **Dynamic:** resampling is taken only when the variance of the importance weights is over the threshold.
- The particles and associated importance weights $\{\mathbf{x}^{(i)}, \tilde{W}^{(i)}\}$ are replaced by the new samples with equal importance weights (i.e. $\tilde{W}^{(i)} = 1/N$).
- Resampling is important because
  - if importance weights are uneven distributed, propagating the trivial weights through the dynamic system is a waste of computing power;
  - when the importance weights are skewed, resampling can provide chances for selecting important samples and rejuvenate the sampler
- Resampling does not necessarily improve the current state estimate because it also introduces extra Monte Carlo variation.
- There are many types of resampling methods.

Gibbs sampling

- It's a particular type of Markov Chain Monte Carlo (MCMC) sampling.
- The Gibbs sampler uses the concept of alternating (marginal) conditional sampling.
- Given an N_x-dimensional state vector $\mathbf{x} = [x_1, x_2, \cdots, x_{N_x}]^T$, we are interested in drawing the samples from the marginal density in the case where joint density is inaccessible or hard to sample.
- Since the conditional density to be sampled is low dimensional, the Gibbs sampler is a nice solution to estimation of hierarchical or structured probabilistic model.

Draw a sample from $\mathbf{x}_0 \sim p(\mathbf{x}_0)$.
for $n = 1$ to M **do**
  for $i = 1$ to N_x **do**
    Draw a sample $x_{i,n} \sim p(x_n | x_{1,n}, \cdots, x_{i-1,n}, x_{i,n-1}, \cdots, x_{N_x,n-1})$
  end for
end for

Illustration of Gibbs sampling

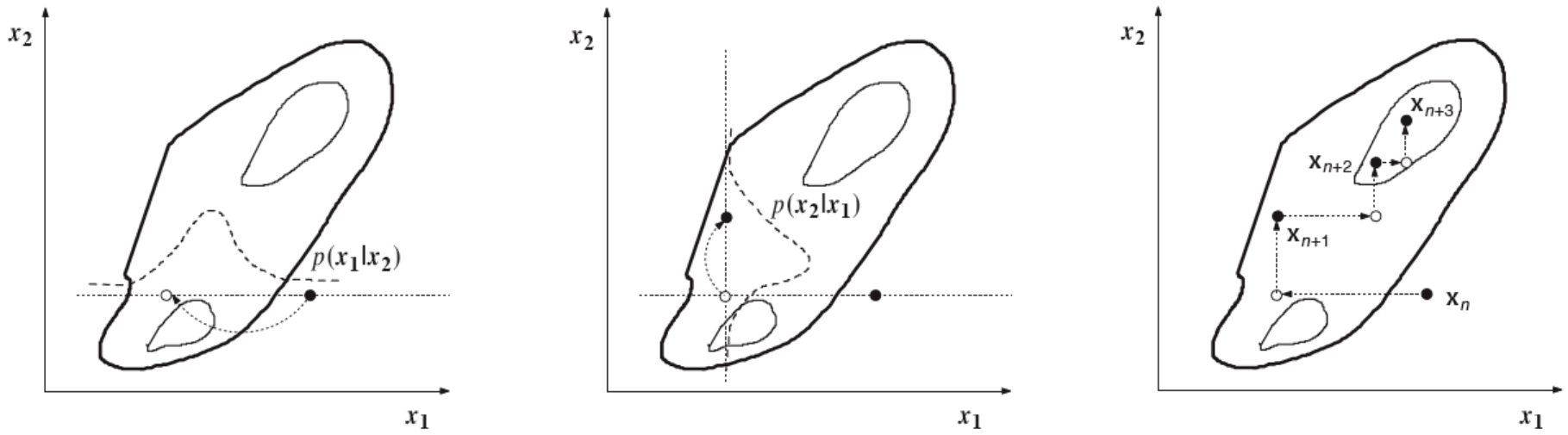

Figure: Gibbs sampling in a two-dimensional space (Chen 03). Left: Starting from state x_n, x_1 is sampled from the conditional pdf $p(x_1|x_{2,n-1})$. Middle: A sample is drawn from the conditional pdf $p(x_2|x_{1,n})$. Right: Four-step iterations in the probability space (contour).

Other sampling strategies

- **Stratified sampling:** distribute the samples evenly (or unevenly according to their respective variance) to the subregions dividing the whole space.
  - Stratified sampling works very well and is efficient in a not-too-high dimension space.
- **Hybrid Monte Carlo:** Metropolis method which uses gradient information to reduce random walk behavior.
  - This is good since the gradient direction might indicate the way to find the state with a higher probability.

Numerical approximations

- Monte-carlo sampling approximation (i.e., particle filter)
- Gaussian/Laplace approximation
- Iterative quadrature
- Multi-grid method and point-mass approximation
- Moment approximation
- Gaussian sum approximation
- Deterministic sampling approximation

Gauss/Laplace approximation

- Gaussian approximation is the simplest method to approximate the numerical integration problem because of its analytic tractability.
- By assuming the posterior as Gaussian, the nonlinear filtering can be taken with the EKF method.
- Laplace approximation method is to approximate the integral of a function $\int f(\mathbf{x})d\mathbf{x}$ by fitting a Gaussian at the maximum $\hat{\mathbf{x}}$ of $f(\mathbf{x})$, and further compute the volume

$$\int f(\mathbf{x})d\mathbf{x} \approx (2\pi)^{N_x/2} f(\hat{\mathbf{x}}) |- \bigtriangledown \bigtriangledown \log f(\mathbf{x})|^{-1/2}$$

- The covariance of the fitted Gaussian is determined by the Hessian matrix of $\log f(\mathbf{x})$ at $\hat{\mathbf{x}}$.

Gauss/Laplace approximation

- Gaussian approximation is the simplest method to approximate the numerical integration problem because of its analytic tractability.
- By assuming the posterior as Gaussian, the nonlinear filtering can be taken with the EKF method.
- Laplace approximation method is to approximate the integral of a function $\int f(\mathbf{x})d\mathbf{x}$ by fitting a Gaussian at the maximum $\hat{\mathbf{x}}$ of $f(\mathbf{x})$, and further compute the volume

$$\int f(\mathbf{x})d\mathbf{x} \approx (2\pi)^{N_x/2} f(\hat{\mathbf{x}}) |- \bigtriangledown \bigtriangledown \log f(\mathbf{x})|^{-1/2}$$

- The covariance of the fitted Gaussian is determined by the Hessian matrix of $\log f(\mathbf{x})$ at $\hat{\mathbf{x}}$.
- It is also used to approximate the posterior distribution with a Gaussian centered a the MAP estimate.
- Works for the unimodal distributions but produces a poor approximation result for multimodal distributions, especially in high-dimensional spaces.

Gauss/Laplace approximation

- Gaussian approximation is the simplest method to approximate the numerical integration problem because of its analytic tractability.
- By assuming the posterior as Gaussian, the nonlinear filtering can be taken with the EKF method.
- Laplace approximation method is to approximate the integral of a function $\int f(\mathbf{x})d\mathbf{x}$ by fitting a Gaussian at the maximum $\hat{\mathbf{x}}$ of $f(\mathbf{x})$, and further compute the volume

$$\int f(\mathbf{x})d\mathbf{x} \approx (2\pi)^{N_x/2} f(\hat{\mathbf{x}}) |- \bigtriangledown \bigtriangledown \log f(\mathbf{x})|^{-1/2}$$

- The covariance of the fitted Gaussian is determined by the Hessian matrix of $\log f(\mathbf{x})$ at $\hat{\mathbf{x}}$.
- It is also used to approximate the posterior distribution with a Gaussian centered a the MAP estimate.
- Works for the unimodal distributions but produces a poor approximation result for multimodal distributions, especially in high-dimensional spaces.

Iterative Quadrature

- Numerical approximation method, which was widely used in computer graphics and physics.
- A finite integral is approximated by a weighted sum of samples of the integrand based on some quadrature formula

$$\int_a^b f(\mathbf{x})p(\mathbf{x})d\mathbf{x} \approx \sum_{k=1}^{m} c_k f(\mathbf{x}_k)$$

  where $p(\mathbf{x})$ is treated as a weighting function, and $\mathbf{x}_k$ is the quadrature point.
- The values $\mathbf{x}_k$ are determined by the weighting function $p(\mathbf{x})$ in the interval $[a, b]$.
- This method can produce a good approximation if the nonlinear function is smooth.

Muti-grid Method and Point-Mass Approximation

- If the state is discrete and finite (or it can be discretized and approximated as finite), grid-based methods can provide a good solution and optimal way to update the filtered density $p(\mathbf{x}_n|\mathbf{y}_{n:0})$.
- If the state space is continuous, we can always discretize the state space into N_z discrete cell states, then a grid-based method can be further used to approximate the posterior density.
- The disadvantage of grid-based method is that it requires the state space cannot be partitioned unevenly to give a great resolution to the state with high density.
- In the point-mass method uses a simple rectangular grid. The density is assumed to be represented by a set of point masses which carry the information about the data.

Moment Approximation

- Moment approximation is targeted at approximating the moments of the density, including mean, covariance, and higher order moments.
- We can empirically use the sample moment to approximate the true moment, namely

$$m_k = E[\mathbf{x}^k] = \int_X \mathbf{x}^k p(\mathbf{x}) d\mathbf{x} = \frac{1}{N} \sum_{i=1}^{N} |\mathbf{x}^{(i)}|^k$$

  where m_k denotes the k-th order moment and $\mathbf{x}^{(i)}$ are the samples from true distribution.
- The computation cost of these approaches are rather prohibitive, especially in highdimensional space.

Gaussian Sum Approximation

- Gaussian sum approximation uses a weighted sum of Gaussian densities to approximate the posterior density (the so-called Gaussian mixture model):

$$p(\mathbf{x}) = \sum_{j=1}^{m} c_j \mathcal{N}(\hat{\mathbf{x}}_f, \Sigma_f)$$

  where the weighting coefficients $c_j > 0$, and $\sum_{j=1}^{m} c_j = 1$

- Any non-Gaussian density can be approximated to some accurate degree by a sufficiently large number of Gaussian mixture densities.
- A mixture of Gaussians admits tractable solution by calculating individual first and second order moments.
- Gaussian sum filter, essentially uses this idea and runs a bank of EKFs in parallel to obtain the suboptimal estimate.

Illustration of numerical approximations

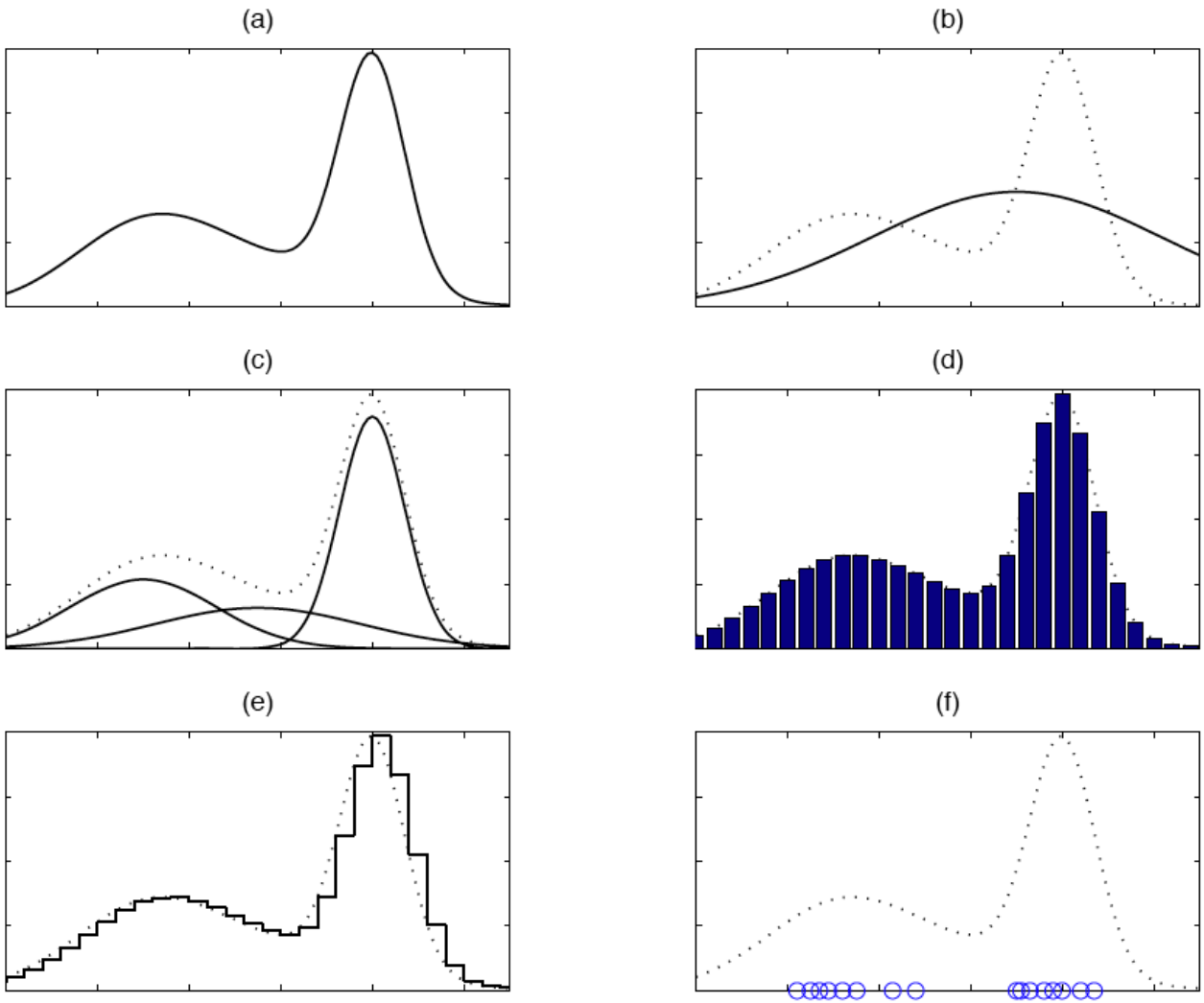

Figure: Illustration of non-Gaussian distribution approximation (Chen 03): (a) true distribution; (b) Gaussian approximation; (c) Gaussian sum approximation; (d) histogram approximation; (e) Riemannian sum (step function) approximation; (f) Monte Carlo sampling approximation.

What have we seen?

We have seen up to now

- Filtering equations
- Monte Carlo sampling
- Other numerical approximation methods

What's next?

- Particle filters

Particle filter: Sequential Monte Carlo estimation

- Now we now how to do numerical approximations. Let's use it!
- Sequential Monte Carlo estimation is a type of recursive Bayesian filter based on Monte Carlo simulation. It is also called **bootstrap filter**.
- The state space is partitioned as many parts, in which the particles are filled according to some probability measure. The higher probability, the denser the particles are concentrated.
- The particle system evolves along the time according to the state equation, with evolving pdf determined by the FPK equation.

Particle filter: Sequential Monte Carlo estimation

- Now we now how to do numerical approximations. Let's use it!
- Sequential Monte Carlo estimation is a type of recursive Bayesian filter based on Monte Carlo simulation. It is also called **bootstrap filter**.
- The state space is partitioned as many parts, in which the particles are filled according to some probability measure. The higher probability, the denser the particles are concentrated.
- The particle system evolves along the time according to the state equation, with evolving pdf determined by the FPK equation.
- Since the pdf can be approximated by the point-mass histogram, by random sampling of the state space, we get a number of particles representing the evolving pdf.
- However, since the posterior density model is unknown or hard to sample, we would rather choose another distribution for the sake of efficient sampling.

Particle filter: Sequential Monte Carlo estimation

- Now we now how to do numerical approximations. Let's use it!
- Sequential Monte Carlo estimation is a type of recursive Bayesian filter based on Monte Carlo simulation. It is also called **bootstrap filter**.
- The state space is partitioned as many parts, in which the particles are filled according to some probability measure. The higher probability, the denser the particles are concentrated.
- The particle system evolves along the time according to the state equation, with evolving pdf determined by the FPK equation.
- Since the pdf can be approximated by the point-mass histogram, by random sampling of the state space, we get a number of particles representing the evolving pdf.
- However, since the posterior density model is unknown or hard to sample, we would rather choose another distribution for the sake of efficient sampling.

Sequential Monte Carlo estimation I

- The posterior distribution or density is empirically represented by a weighted sum of N samples drawn from the posterior distribution

$$p(\mathbf{x}_n|\mathbf{y}_{n:0}) \approx \frac{1}{N}\sum_{i=1}^{N}\delta(\mathbf{x}_n - \mathbf{x}_n^{(i)}) \equiv \hat{p}(\mathbf{x}_n|\mathbf{y}_{n:0})$$

  where $\mathbf{x}_n^{(i)}$ are assumed to be i.i.d. drawn from $p(\mathbf{x}_n|\mathbf{y}_{n:0})$.

- By this approximation, we can estimate the mean of a nonlinear function

$$\begin{aligned} E[f(\mathbf{x}_n)] &\approx \int f(\mathbf{x}_n)\hat{p}(\mathbf{x}_n|\mathbf{y}_{n:0})d\mathbf{x}_n \\ &= \frac{1}{N}\sum_{i=1}^{N}\int f(\mathbf{x}_n)\delta(\mathbf{x}_n - \mathbf{x}_n^{(i)})d\mathbf{x}_n \\ &= \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_n^{(i)}) \equiv \hat{f}_N(\mathbf{x}) \end{aligned}$$

Sequential Monte Carlo estimation I

- The posterior distribution or density is empirically represented by a weighted sum of N samples drawn from the posterior distribution

$$p(\mathbf{x}_n|\mathbf{y}_{n:0}) \approx \frac{1}{N}\sum_{i=1}^{N}\delta(\mathbf{x}_n - \mathbf{x}_n^{(i)}) \equiv \hat{p}(\mathbf{x}_n|\mathbf{y}_{n:0})$$

  where $\mathbf{x}_n^{(i)}$ are assumed to be i.i.d. drawn from $p(\mathbf{x}_n|\mathbf{y}_{n:0})$.

- By this approximation, we can estimate the mean of a nonlinear function

$$\begin{aligned} E[f(\mathbf{x}_n)] &\approx \int f(\mathbf{x}_n)\hat{p}(\mathbf{x}_n|\mathbf{y}_{n:0})d\mathbf{x}_n \\ &= \frac{1}{N}\sum_{i=1}^{N}\int f(\mathbf{x}_n)\delta(\mathbf{x}_n - \mathbf{x}_n^{(i)})d\mathbf{x}_n \\ &= \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_n^{(i)}) \equiv \hat{f}_N(\mathbf{x}) \end{aligned}$$

Sequential Monte Carlo estimation II

- It is usually impossible to sample from the true posterior, it is common to sample from the so-called *proposal distribution* $q(\mathbf{x}_n|\mathbf{y}_{n:0})$. Let's define

$$W_n(\mathbf{x}_n) = \frac{p(\mathbf{y}_{n:0}|\mathbf{x}_n)p(\mathbf{x}_n)}{q(\mathbf{x}_n|\mathbf{y}_{n:0})}$$

- We can then write

$$E[f(\mathbf{x}_n)] = \int f(\mathbf{x}_n)\frac{p(\mathbf{x}_n|\mathbf{y}_{n:0})}{q(\mathbf{x}_n|\mathbf{y}_{n:0})}q(\mathbf{x}_n|\mathbf{y}_{n:0})d\mathbf{x}_n$$

Sequential Monte Carlo estimation II

- It is usually impossible to sample from the true posterior, it is common to sample from the so-called *proposal distribution* $q(\mathbf{x}_n|\mathbf{y}_{n:0})$. Let's define

$$W_n(\mathbf{x}_n) = \frac{p(\mathbf{y}_{n:0}|\mathbf{x}_n)p(\mathbf{x}_n)}{q(\mathbf{x}_n|\mathbf{y}_{n:0})}$$

- We can then write

$$\begin{aligned} E[f(\mathbf{x}_n)] &= \int f(\mathbf{x}_n)\frac{p(\mathbf{x}_n|\mathbf{y}_{n:0})}{q(\mathbf{x}_n|\mathbf{y}_{n:0})}q(\mathbf{x}_n|\mathbf{y}_{n:0})d\mathbf{x}_n \\ &= \int f(\mathbf{x}_n)\frac{W_n(\mathbf{x}_n)}{p(\mathbf{y}_{n:0})}q(\mathbf{x}_n|\mathbf{y}_{n:0})d\mathbf{x}_n \end{aligned}$$

Sequential Monte Carlo estimation II

- It is usually impossible to sample from the true posterior, it is common to sample from the so-called *proposal distribution* $q(\mathbf{x}_n|\mathbf{y}_{n:0})$. Let's define

$$W_n(\mathbf{x}_n) = \frac{p(\mathbf{y}_{n:0}|\mathbf{x}_n)p(\mathbf{x}_n)}{q(\mathbf{x}_n|\mathbf{y}_{n:0})}$$

- We can then write

$$\begin{aligned} E[f(\mathbf{x}_n)] &= \int f(\mathbf{x}_n)\frac{p(\mathbf{x}_n|\mathbf{y}_{n:0})}{q(\mathbf{x}_n|\mathbf{y}_{n:0})}q(\mathbf{x}_n|\mathbf{y}_{n:0})d\mathbf{x}_n \\ &= \int f(\mathbf{x}_n)\frac{W_n(\mathbf{x}_n)}{p(\mathbf{y}_{n:0})}q(\mathbf{x}_n|\mathbf{y}_{n:0})d\mathbf{x}_n \\ &= \frac{\int f(\mathbf{x}_n)W_n(\mathbf{x}_n)q(\mathbf{x}_n|\mathbf{y}_{n:0})d\mathbf{x}_n}{\int p(\mathbf{y}_{n:0}|\mathbf{x}_n)p(\mathbf{x}_n)d\mathbf{x}_n} \end{aligned}$$

Sequential Monte Carlo estimation II

- It is usually impossible to sample from the true posterior, it is common to sample from the so-called *proposal distribution* $q(\mathbf{x}_n|\mathbf{y}_{n:0})$. Let's define

$$W_n(\mathbf{x}_n) = \frac{p(\mathbf{y}_{n:0}|\mathbf{x}_n)p(\mathbf{x}_n)}{q(\mathbf{x}_n|\mathbf{y}_{n:0})}$$

- We can then write

$$\begin{aligned}
E[f(\mathbf{x}_n)] &= \int f(\mathbf{x}_n)\frac{p(\mathbf{x}_n|\mathbf{y}_{n:0})}{q(\mathbf{x}_n|\mathbf{y}_{n:0})}q(\mathbf{x}_n|\mathbf{y}_{n:0})d\mathbf{x}_n \\
&= \int f(\mathbf{x}_n)\frac{W_n(\mathbf{x}_n)}{p(\mathbf{y}_{n:0})}q(\mathbf{x}_n|\mathbf{y}_{n:0})d\mathbf{x}_n \\
&= \frac{\int f(\mathbf{x}_n)W_n(\mathbf{x}_n)q(\mathbf{x}_n|\mathbf{y}_{n:0})d\mathbf{x}_n}{\int p(\mathbf{y}_{n:0}|\mathbf{x}_n)p(\mathbf{x}_n)d\mathbf{x}_n} \\
&= \frac{\int f(\mathbf{x}_n)W_n(\mathbf{x}_n)q(\mathbf{x}_n|\mathbf{y}_{n:0})d\mathbf{x}_n}{\int W_n(\mathbf{x}_n)q(\mathbf{x}_n|\mathbf{y}_{n:0})d\mathbf{x}_n}
\end{aligned}$$

Sequential Monte Carlo estimation II

- It is usually impossible to sample from the true posterior, it is common to sample from the so-called *proposal distribution* $q(\mathbf{x}_n|\mathbf{y}_{n:0})$. Let's define

$$W_n(\mathbf{x}_n) = \frac{p(\mathbf{y}_{n:0}|\mathbf{x}_n)p(\mathbf{x}_n)}{q(\mathbf{x}_n|\mathbf{y}_{n:0})}$$

- We can then write

$$\begin{aligned}
E[f(\mathbf{x}_n)] &= \int f(\mathbf{x}_n)\frac{p(\mathbf{x}_n|\mathbf{y}_{n:0})}{q(\mathbf{x}_n|\mathbf{y}_{n:0})}q(\mathbf{x}_n|\mathbf{y}_{n:0})d\mathbf{x}_n \\
&= \int f(\mathbf{x}_n)\frac{W_n(\mathbf{x}_n)}{p(\mathbf{y}_{n:0})}q(\mathbf{x}_n|\mathbf{y}_{n:0})d\mathbf{x}_n \\
&= \frac{\int f(\mathbf{x}_n)W_n(\mathbf{x}_n)q(\mathbf{x}_n|\mathbf{y}_{n:0})d\mathbf{x}_n}{\int p(\mathbf{y}_{n:0}|\mathbf{x}_n)p(\mathbf{x}_n)d\mathbf{x}_n} \\
&= \frac{\int f(\mathbf{x}_n)W_n(\mathbf{x}_n)q(\mathbf{x}_n|\mathbf{y}_{n:0})d\mathbf{x}_n}{\int W_n(\mathbf{x}_n)q(\mathbf{x}_n|\mathbf{y}_{n:0})d\mathbf{x}_n} \\
&= \frac{E_{q(\mathbf{x}_n|\mathbf{y}_{n:0})}[W_n(\mathbf{x}_n)f(\mathbf{x}_n)]}{E_{q(\mathbf{x}_n|\mathbf{y}_{n:0})}[W_n(\mathbf{x}_n)]}
\end{aligned}$$

Sequential Monte Carlo estimation II

- It is usually impossible to sample from the true posterior, it is common to sample from the so-called *proposal distribution* $q(\mathbf{x}_n|\mathbf{y}_{n:0})$. Let's define

$$W_n(\mathbf{x}_n) = \frac{p(\mathbf{y}_{n:0}|\mathbf{x}_n)p(\mathbf{x}_n)}{q(\mathbf{x}_n|\mathbf{y}_{n:0})}$$

- We can then write

$$\begin{aligned}
E[f(\mathbf{x}_n)] &= \int f(\mathbf{x}_n)\frac{p(\mathbf{x}_n|\mathbf{y}_{n:0})}{q(\mathbf{x}_n|\mathbf{y}_{n:0})}q(\mathbf{x}_n|\mathbf{y}_{n:0})d\mathbf{x}_n \\
&= \int f(\mathbf{x}_n)\frac{W_n(\mathbf{x}_n)}{p(\mathbf{y}_{n:0})}q(\mathbf{x}_n|\mathbf{y}_{n:0})d\mathbf{x}_n \\
&= \frac{\int f(\mathbf{x}_n)W_n(\mathbf{x}_n)q(\mathbf{x}_n|\mathbf{y}_{n:0})d\mathbf{x}_n}{\int p(\mathbf{y}_{n:0}|\mathbf{x}_n)p(\mathbf{x}_n)d\mathbf{x}_n} \\
&= \frac{\int f(\mathbf{x}_n)W_n(\mathbf{x}_n)q(\mathbf{x}_n|\mathbf{y}_{n:0})d\mathbf{x}_n}{\int W_n(\mathbf{x}_n)q(\mathbf{x}_n|\mathbf{y}_{n:0})d\mathbf{x}_n} \\
&= \frac{E_{q(\mathbf{x}_n|\mathbf{y}_{n:0})}[W_n(\mathbf{x}_n)f(\mathbf{x}_n)]}{E_{q(\mathbf{x}_n|\mathbf{y}_{n:0})}[W_n(\mathbf{x}_n)]}
\end{aligned}$$

Sequential Monte Carlo estimation III

- We have written

$$E[f(\mathbf{x}_n)] = \frac{E_{q(\mathbf{x}_n|\mathbf{y}_{n:0})}[W_n(\mathbf{x}_n)f(\mathbf{x}_n)]}{E_{q(\mathbf{x}_n|\mathbf{y}_{n:0})}[W_n(\mathbf{x}_n)]}$$

- By drawing the i.i.d. samples $\{\mathbf{x}_n^{(i)}\}$ from $q(\mathbf{x}_n|\mathbf{y}_{n:0})$, we can approximate

$$E[f(\mathbf{x}_n)] \approx \frac{\frac{1}{N}\sum_{i=1}^{N} W_n(\mathbf{x}_n^{(i)})f(\mathbf{x}_n^{(i)})}{\frac{1}{N}\sum_{i=1}^{N} W_n(\mathbf{x}_n^{(i)})} = \sum_{i=1}^{N} \tilde{W}(\mathbf{x}_n^{(i)})f(\mathbf{x}_n^{(i)}) \equiv \hat{f}(\mathbf{x})$$

where the normalized weights are defined as

$$\tilde{W}(\mathbf{x}_n^{(i)}) = \frac{W_n(\mathbf{x}_n^{(i)})}{\sum_{i=1}^{N} W_n(\mathbf{x}_n^{(i)})}$$

Sequential Monte Carlo estimation IV

- Suppose now that the proposal distribution factorizes

$$q(\mathbf{x}_{n:0}|\mathbf{y}_{n:0}) = q(\mathbf{x}_0)\prod_{t=1}^{n} q(\mathbf{x}_t|\mathbf{x}_{t-1:0}, \mathbf{y}_{t:0})$$

- As before the posterior can be written as

$$p(\mathbf{x}_{n:0}|\mathbf{y}_{n:0}) = p(\mathbf{x}_{n-1:0}|\mathbf{y}_{n-1:0})\frac{p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{n-1:0})}{p(\mathbf{y}_n|\mathbf{y}_{n-1:0})}$$

- We can then create a recursive rule to update the weights

$$\begin{aligned} W_n^{(i)} &= \frac{p(\mathbf{x}_{n:0}^{(i)}|\mathbf{y}_{n:0})}{q(\mathbf{x}_{n:0}^{(i)}|\mathbf{y}_{n:0})} \\ &\propto \frac{p(\mathbf{y}_n|\mathbf{x}_n^{(i)})p(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)})p(\mathbf{x}_{n-1:0}^{(i)}|\mathbf{y}_{n-1:0})}{q(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1:0}^{(i)}, \mathbf{y}_{n:0})q(\mathbf{x}_{n-1:0}^{(i)}|\mathbf{y}_{n-1:0})} \\ &= W_{n-1}^{(i)}\frac{p(\mathbf{y}_n|\mathbf{x}_n^{(i)})p(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)})}{q(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1:0}^{(i)}, \mathbf{y}_{n:0})} \end{aligned}$$

Types of filters

Depending on the type of sampling use we have different types of filters

- Sequential Importance sampling (SIS) filter
- SIR filter
- Auxiliary particle filter (APF)
- Rejection particle filter
- MCMC particle filter
- etc.

Sequential Importance sampling (SIS) filter I

- We are more interested in the current filtered estimate $p(\mathbf{x}_n|\mathbf{y}_{n:0})$ than $p(\mathbf{x}_{n:0}|\mathbf{y}_{n:0})$.
- Let's assume that $q(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1:0}^{(i)}, \mathbf{y}_{n:0}) = q(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1:0}^{(i)}, \mathbf{y}_n)$ then we can write

$$W_n^{(i)} = W_{n-1}^{(i)} \frac{p(\mathbf{y}_n|\mathbf{x}_n^{(i)})p(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)})}{q(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1:0}^{(i)}, \mathbf{y}_n)}$$

- The problem of the SIS filter is that the distribution of the importance weights becomes more and more skewed as time increases.
- After some iterations, only very few particles have non-zero importance weights. This is often called **weight degeneracy** or **sample impoverishment**.

Sequential Importance sampling (SIS) filter II

- A solution is to multiply the particles with high normalized importance weights, and discard the particles with low normalized importance weights, which can be be done in the resampling step.
- A suggested measure for degeneracy is the so-called **effective sample size**

$$N_{eff} = \frac{N}{E_{q(\cdot|\mathbf{y}_{n:0})}[(\tilde{W}(\mathbf{x}_{n:0}))^2]} \leq N$$

- In practice this cannot be computed, so we approximate

$$N_{eff} \approx \frac{1}{\sum_{i=1}^{N}(\tilde{W}(\mathbf{x}_{n:0}))^2}$$

- When N_{eff} is below a threshold P, then resampling is performed.
- N_{eff} can be also used to combine rejection and importance sampling

SIS particle filter with resampling

for $n = 0, \cdots, T$ **do**
  for $i = 1, \cdots, N$ **do**
    Draw samples $\mathbf{x}_n^{(i)} \sim q(\mathbf{x}_n | \mathbf{x}_{n-1:0}^{(i)}, \mathbf{y}_{n:0})$
    Set $\mathbf{x}_{n:0}^{(i)} = \{\mathbf{x}_{n-1:0}^{(i)}, \mathbf{x}_n^{(i)}\}$
  end for
  for $i = 1, \cdots, N$ **do**
    Calculate weights $W_n^{(i)} = W_{n-1}^{(i)} \frac{p(\mathbf{y}_n|\mathbf{x}_n^{(i)}) p(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)})}{q(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1:0}^{(i)}, \mathbf{y}_n)}$
  end for
  for $i = 1, \cdots, N$ **do**
    Normalize the weights $\tilde{W}(\mathbf{x}^{(i)}) = \frac{W(\mathbf{x}^{(i)})}{\sum_{i=1}^{N} W(\mathbf{x}^{(i)})}$
  end for
  Compute $\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N} (\tilde{W}(\mathbf{x}_{n:0}))^2}$
  if $\hat{N}_{eff} < P$ **then**
    Generate new $\{\mathbf{x}_n^{(j)}\}$ by resampling with replacement N times from $\{\mathbf{x}_{n:0}^{(i)}\}$ with probability $P(\mathbf{x}_{n:0}^{(j)} = \mathbf{x}_{n:0}^{(i)}) = \tilde{W}_{n:0}^{(i)}$.
    Reset the weights $\tilde{W}_n^{(i)} = \frac{1}{N}$
  end if
end for

Bootstrap/SIR filter

- The key idea of SIR filter is to introduce the resampling step as in the SIR sampling.
- Resampling does not really prevent the weight degeneracy problem, it just saves further calculation time by discarding the particles associated with insignificant weights.
- It artificially concealing the impoverishment by replacing the high important weights with many replicates of particles, thereby introducing high correlation between particles.

SIR filter using transition prior as proposal distribution

for $i = 1, \cdots, N$ **do**
  Sample $\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0)$
  Compute $W_0^{(i)} = \frac{1}{N}$
end for
for $n = 0, \cdots, T$ **do**
  for $i = 1, \cdots, N$ **do**
    Importance sampling $\hat{\mathbf{x}}_n^{(i)} \sim p(\mathbf{x}_n | \mathbf{x}_{n-1}^{(i)})$
  end for
  Set $\hat{\mathbf{x}}_{n:0}^{(i)} = \{\mathbf{x}_{n-1:0}^{(i)}, \hat{\mathbf{x}}_n^{(i)}\}$
  for $i = 1, \cdots, N$ **do**
    Weight update $W_n^{(i)} = p(\mathbf{y}_n | \hat{\mathbf{x}}_n^{(i)})$
  end for
  for $i = 1, \cdots, N$ **do**
    Normalize weights $\tilde{W}(\mathbf{x}^{(i)}) = \frac{W(\mathbf{x}^{(i)})}{\sum_{i=1}^{N} W(\mathbf{x}^{(i)})}$
  end for
  Resampling: Generate N new particles $\mathbf{x}_n^{(i)}$ from the set $\{\hat{\mathbf{x}}_n^{(i)}\}$ according to $\tilde{W}_n^{(i)}$.
end for

Illustration of a generic particle filter

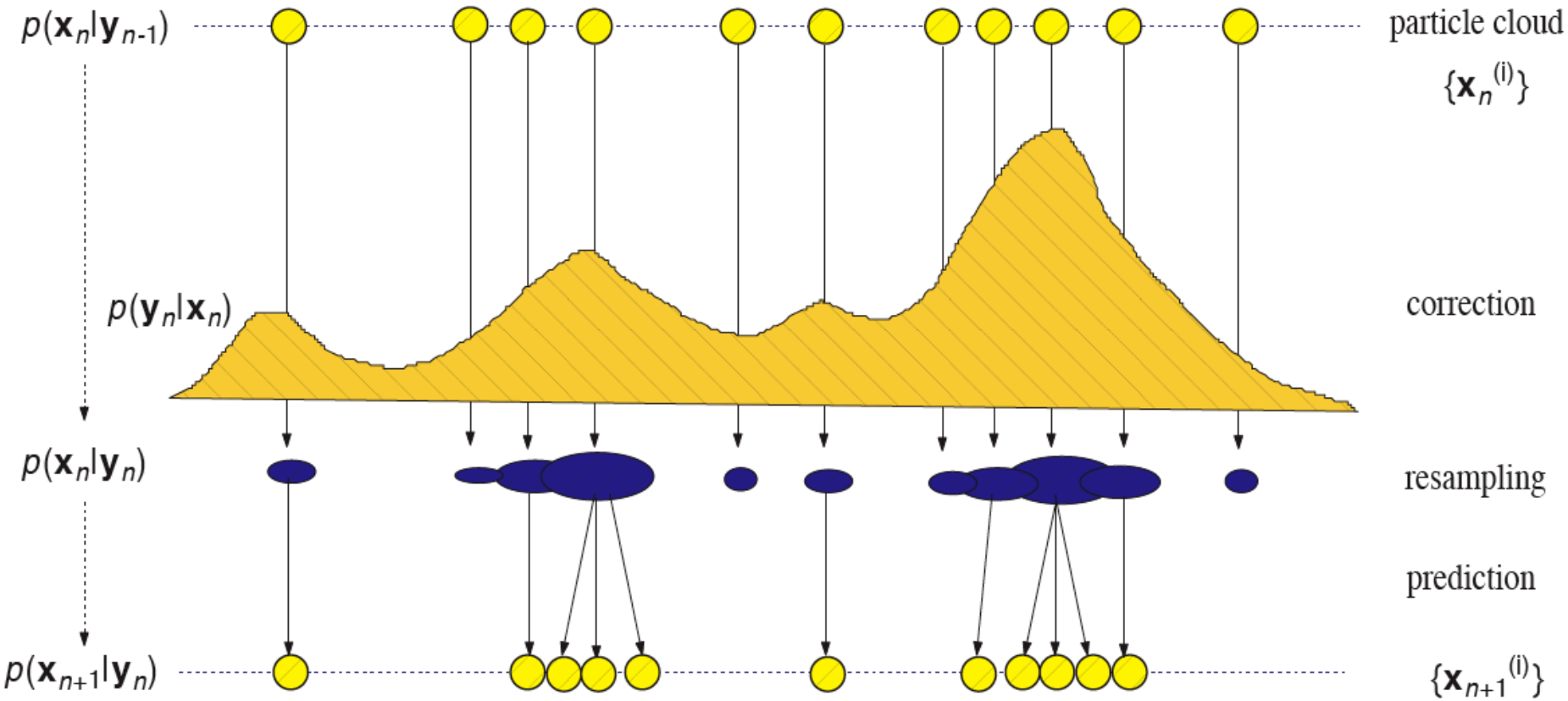

Figure: Particle filter with importance sampling and resampling (Chen 03)

Remarks on SIS and SIR filters

- In the SIR filter the resampling is always performed.
- In the SIS filter, importance weights are calculated sequentially, resampling is only taken whenever needed; SIS filter is less computationally expensive.
- The choice of proposal distributions in SIS and SIR filters plays an crucial role in their final performance.
- Normally the posterior estimate (and its relevant statistics) should be calculated before resampling.
- In the resampling stage, the new importance weights of the surviving particles are not necessarily reset to $1/N$, but rather more clever strategies.
- To alleviate the sample degeneracy in SIS filter, we can change

$$W_n = W_{n-1}^{\alpha} \frac{p(\mathbf{y}_n|\mathbf{x}_n^{(i)})p(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)})}{q(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1:0}^{(i)}, \mathbf{y}_n)}$$

where $0 < \alpha < 1$ is the annealing factor that controls the impact of previous importance weights.

Popular CONDENSATION

Figure: CONDENSATION

Popular CONDENSATION

Figure: Head tracking

Figure: Leaf tracking

Popular CONDENSATION

Figure: Hand tracking

Figure: Hand drawing

Popular CONDENSATION

Figure: Hand tracking

Figure: Interactive applications

More?

- If you want to learn more, look at the additional material.
- Otherwise, do the research project on this topic!
- Next week we will do human pose estimation
- Let's do some exercises now!