Master in Computer Vision Barcelona

Module: Video Analysis

Lecture 5: Bayesian tracking (II)

Lecturer: Ramon Morros

Slides: David Varas

Some slides adapted from: Introduction to tracking by K. Smith at EPFL, Lausanne

Presentation outline

- Introduction

- Monte Carlo Methods
  - Sampling

- Motivation: Particle Filters
  - Importance sampling

- Particle Filters
  - Generic Particle Filter
  - SIR particle filter

- Summary

- References

Introduction

NON-LINEAR BAYESIAN TRACKING

- **No assumptions**
  - **Functions** that define the propagation of the system state between consecutive time instants are **not necessary known** (In contrast with the Kalman Filter).
  - **No restrictions** to these functions (In contrast with the Kalman Filter).
  - **No particular case** of the noise (In general not Gaussian).

$$\begin{cases} x_n = \phi\,(x_{n-1}, v_{n-1}) \\ z_n = \psi\,(x_n, w_n) \end{cases}$$

- **Iteratively estimation** of the posterior pdf $P(X_t \mid z_0, \ldots, z_t)$
  - Arbitrary
  - Multidimensional
  - Time varying

→ **Monte Carlo Methods**

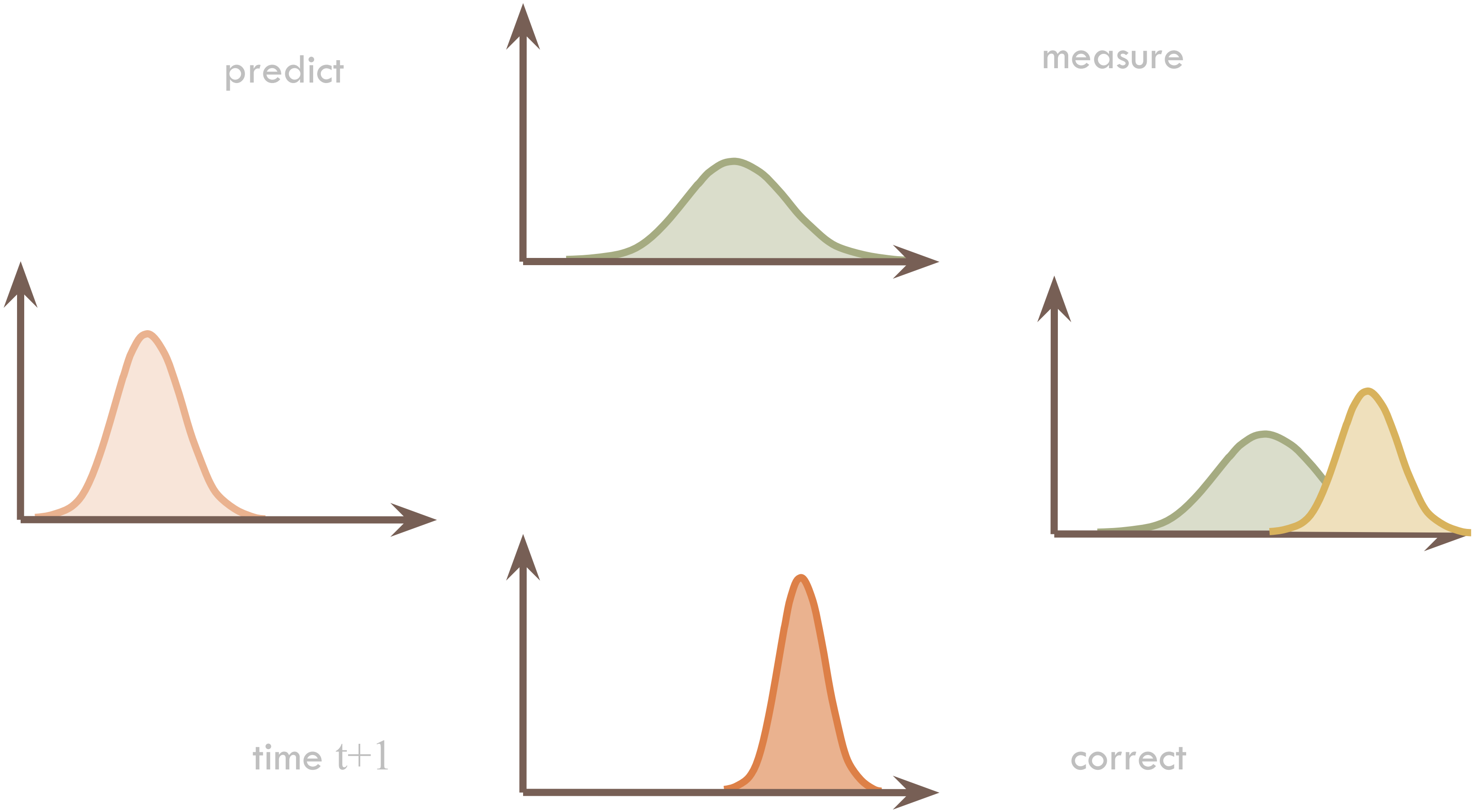
Introduction

Linear!

$x_n = \phi\,(\,x_{n-1},\, v_{n-1}\,)$

$z_n = \psi\,(\,x_n,\, w_n\,)$

- **Gaussian densities** → Kalman filter



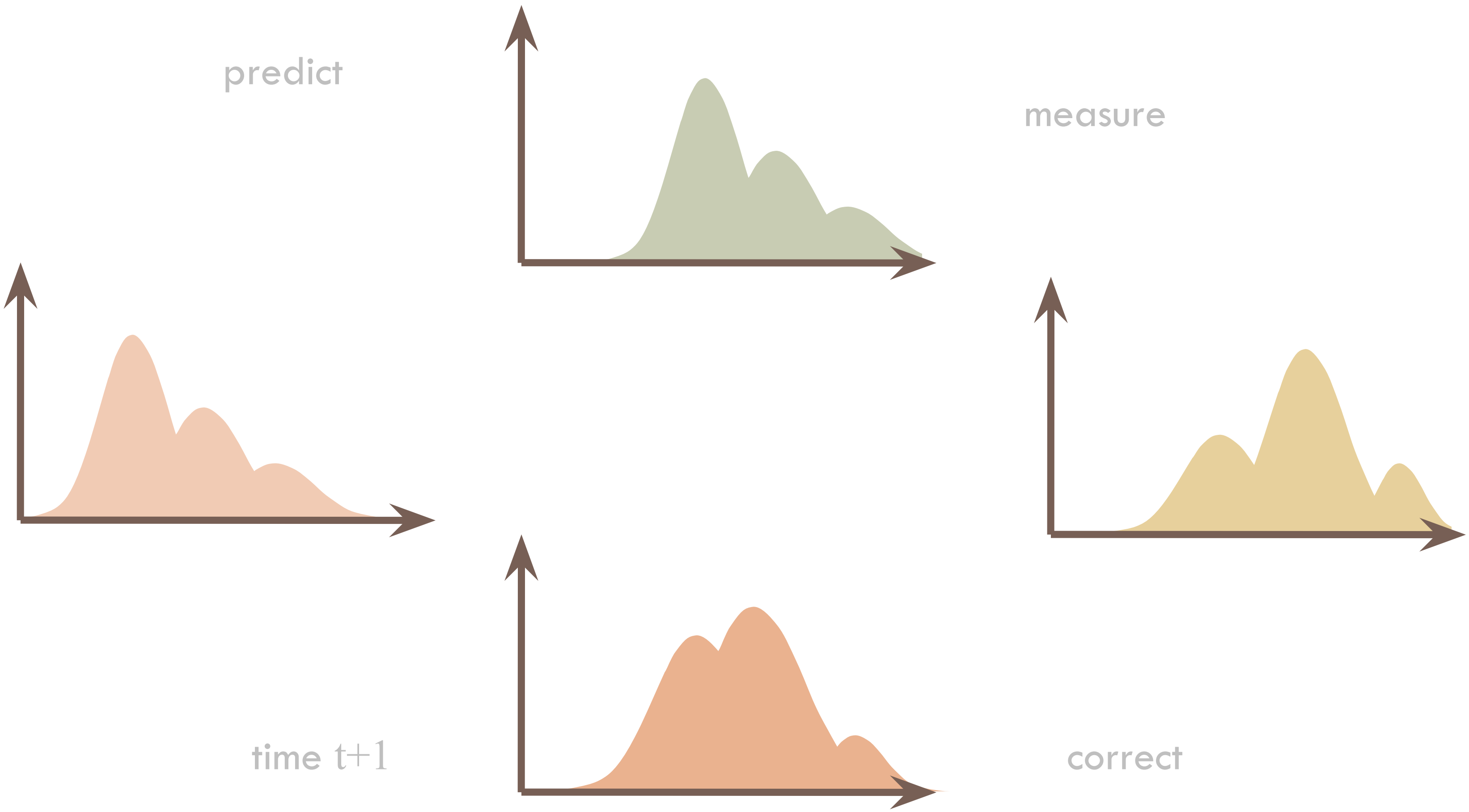
Introduction

Non-linear, maybe unknown

$$x_n = \phi\,(\,x_{n-1},\, v_{n-1}\,)$$

$$z_n = \psi\,(\,x_n,\, w_n\,)$$

- **General densities** → Particle filter



Presentation outline

- Introduction

- **Monte Carlo Methods**
  - Sampling

- Motivation: Particle Filters
  - Importance sampling

- Particle Filters
  - Generic Particle Filter
  - SIR particle filter

- Summary

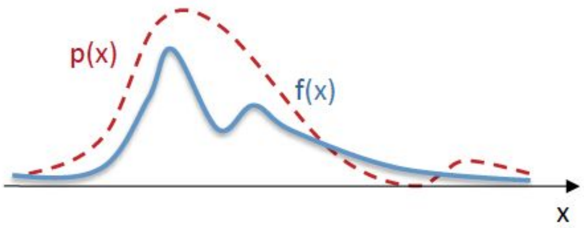
- References

Monte Carlo Methods

MOTIVATION

- In the context of **signal processing**, the estimation of **expectations** is paramount and complex:

$$E\left[f(x)\right] = \int f(x)p(x)\,dx \quad \textbf{(1)}$$



is the expectation of *f(x)* knowing that x is a random variable generated by *p(x)*

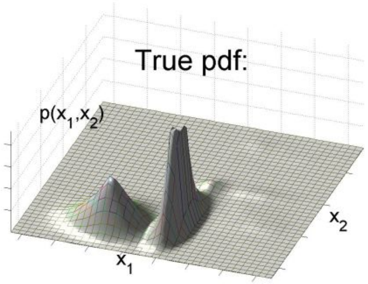
Monte Carlo Methods

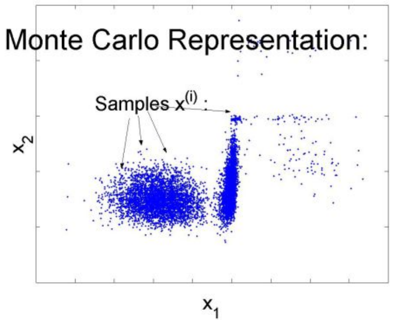
MONTE CARLO METHODS

- **Monte Carlo Methods:** sampling methods that use a set of random samples to estimate highly complex deterministic results.

- **Stochastic vs Deterministic Methods**
  - Deterministic: Same results are obtained every time that simulations are run.
  - Stochastic: Different results at each simulation because there is a randomness element in it.

- **Applications**
  - Physical sciences
  - Engineering
  - Computational biology
  - Computer graphics
  - Statistics
  - Design and visuals
  - Finance and business





- Introduced to the CV community in 1998
  - **M. Isard and A. Blake**, [CONDENSATION – Conditional Density Propagation for Visual Tracking](), IJCV 1998.

Monte Carlo Methods

MONTE CARLO METHODS

In the Monte Carlo Methods, we are concerned with estimating the properties of some complex probability distribution p(x), e.g. the expectation:

$$E\left[f(x)\right] = \int f(x)p(x)\,dx$$

Where *f(·)* is some useful function for estimation.

In cases where this cannot be achieved analytically, the approximation problem can be tackled indirectly, as it is often possible to generate random samples from p(x), i.e. by representing the distribution as a collection of random samples:

$$x^i, i = 1,\ldots N,$$ *for large N*

Π approximation

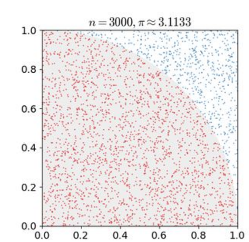
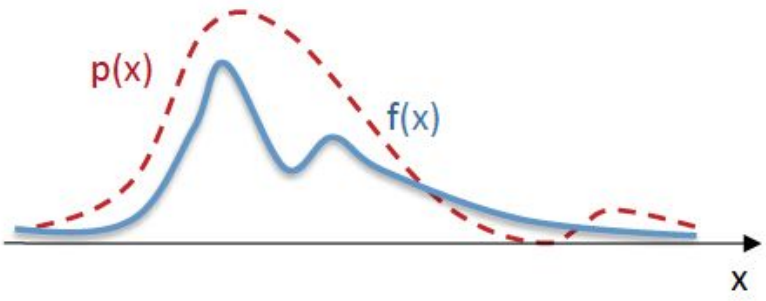


Image: https://en.wikipedia.org/wiki/Monte_Carlo_method

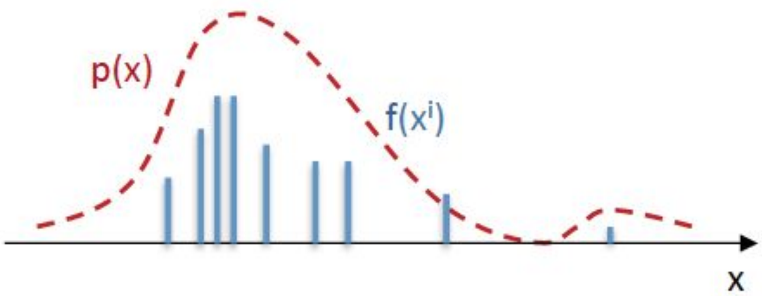
Monte Carlo Methods

MONTE CARLO METHODS

Let's define a function *f(·)* over an iid process *p(x)*



If $x^1, \dots, x^n \sim p(x)$, we can obtain (draw) N samples of the function:



Monte Carlo Methods

MONTE CARLO METHODS

Monte Carlo methods: sampling methods that use a set of random samples to estimate highly complex deterministic results

$$E[X] = E[f(x)] = \int f(x)p(x)dx \Rightarrow E[X] \approx \frac{1}{N}\sum_{i=1}^{N} f(x^i)$$

Unbiased

Consistent

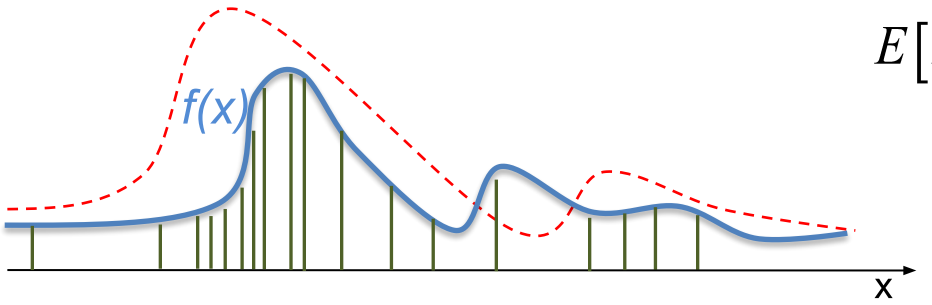
Variance tends to zero as N grows

How are random samples selected?

- Uniform sampling
- Importance sampling
- Rejection sampling
- …

Monte Carlo Methods

SAMPLING: MOTIVATION



$$E[X] = \int f(x)p(x)\,dx$$

- If we can sample p(x), we can use the Monte Carlo estimator: $E[X] \approx \frac{1}{N}\sum_{i=1}^{N} f(x^i)$
- But, **can we sample p(x)??**
  - In most cases, p(x) is unknown →We can not sample it ☹
  - Sometimes, the shape of the distribution is know (up to a constant):

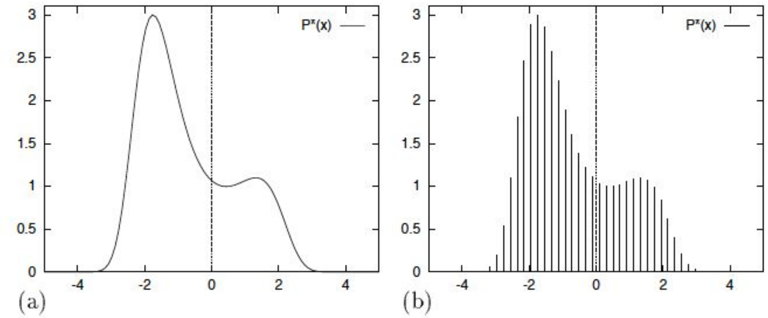
$$p(x) \propto \tilde{p}(x)$$

Monte Carlo Methods

SAMPLING: MOTIVATION

- In the best cases, we know the distribution up to a normalizing constant.

$$p(x) = \frac{\tilde{p}(x)}{Z}$$



$$P^*(x) = \exp\left[0.4(x-0.4)^2 - 0.08x^4\right], \quad x \in (-\infty, \infty)$$

- $\tilde{p}(x)$ can be evaluated!

- Problems:
  - Computing the normalizing constant implies high computational complexity in high dimensional spaces

$$Z = \int \tilde{p}(x)\,dx \qquad \textbf{(3)}$$

  - Even if we can plot *p*(x)*, we do not know how to draw samples from it (There are only a few high-dimensional densities from which it is easy to draw samples)

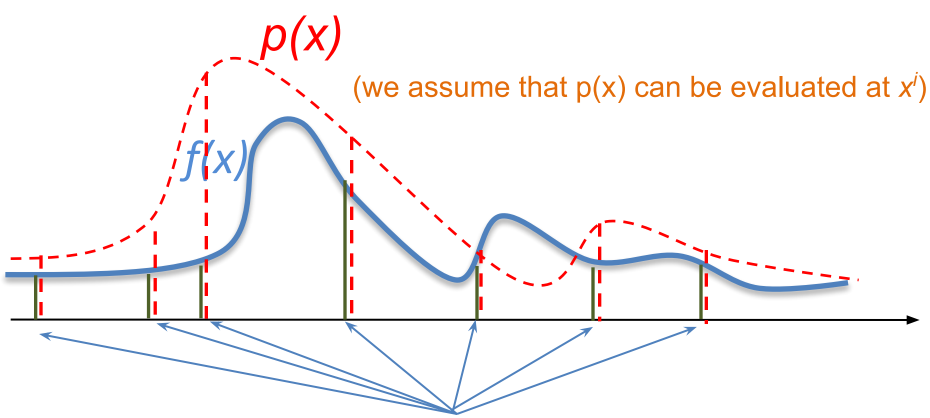
Monte Carlo Methods

UNIFORM SAMPLING

- Draw random samples $\{x^i\}_{i=1,\dots,N}$ uniformly from the space X and estimate objective function.

$$E[X] = \int f(x)p(x)\,dx \quad \Rightarrow \quad E[X] = \frac{1}{N}\sum_{i=1}^{N} f(x^i)p(x^i)$$

- Good estimator of the function? → If N is small, the expectation will not be accurately computed!



Samples drawn from an uniform distribution

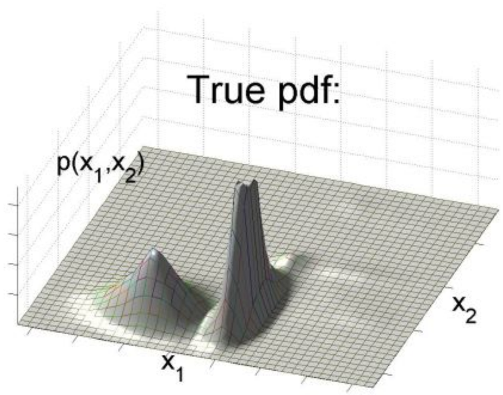
Monte Carlo Methods

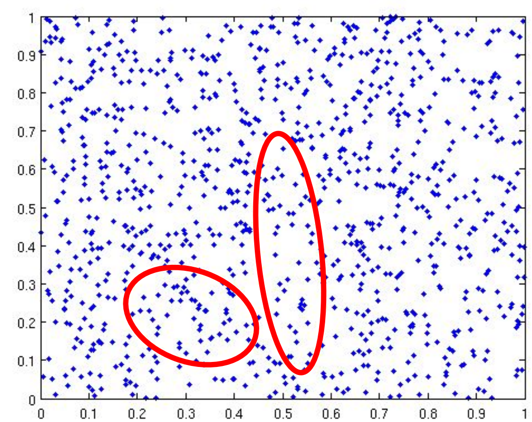
UNIFORM SAMPLING

- Draw random samples $\{x^i\}_{i=1,\dots,N}$ uniformly from the space *X* and estimate objective function.

$$E[X] = \int f(x)p(x)\,dx \quad \Longrightarrow \quad E[X] = \frac{1}{N}\sum_{i=1}^{N} f(x^i)p(x^i)$$

- Good estimator of the function?





Monte Carlo Methods

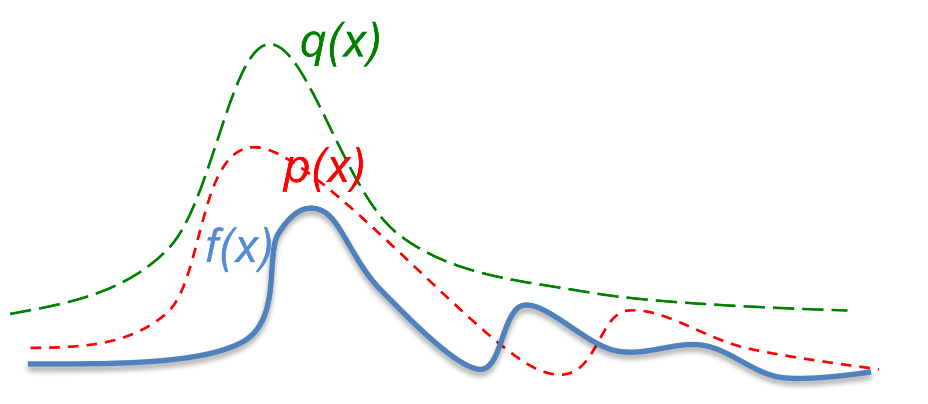
IMPORTANCE SAMPLING

- Most times we can not sample *p(x)* → can we generate samples from other distributions to improve results?

Let $x^i \sim q(x), i=1,\ldots,N$ be samples generated from a proposal $q(\cdot)$ called an ***importance density***.

As we sampled from the 'wrong' distribution, we introduce weights in the estimation:

$$E[X] \approx \frac{1}{N}\sum_{i=1}^{N} w^i f\left(x^i\right) \quad \textbf{(4)}$$



where:

$$\mathrm{w}^{\mathrm{i}} = \frac{p(\mathrm{x}^{\mathrm{i}})}{\mathrm{q}(\mathrm{x}^{\mathrm{i}})} \quad \textbf{(5)}$$

is the **importance weight** of the i_{th} particle

Monte Carlo Methods

IMPORTANCE SAMPLING WITHOUT NORMALIZATION

Suppose *p(x)* is a probability density function from which it is difficult to draw samples. Let $x^i \sim q(x), i=1,\ldots, N$ be samples generated from a proposal $q(\cdot)$ called an ***importance density***.

$$E[X] \approx \frac{1}{N}\sum_{i=1}^{N} w^i f(x_i) \qquad \text{with} \qquad w^i = \frac{p(x^i)}{q(x^i)}$$

Implications:

- Capacity to generate samples from q(x) ✔
- Capacity to evaluate f(x) ✔
- Capacity to evaluate p(x) ✗
- Capacity to evaluate q(x) ✗

Monte Carlo Methods

IMPORTANCE SAMPLING WITHOUT NORMALIZATION

Suppose *p(x)* is a probability density function **that is known up to a normalization factor** and from which it is difficult to draw samples.

Let $x^i \sim q(x)$, $i=1,\ldots, Ns$ be samples generated from a proposal $q(\cdot)$, **known up to a normalization factor***,* called an ***importance density***.

$$p(x) = \frac{\tilde{p}(x)}{z_p} \qquad q(x) = \frac{\tilde{q}(x)}{z_q}$$

$$\int \tilde{p}(x)\,dx = z_p \qquad \int \tilde{q}(x)\,dx = z_q$$

High computational complexity if dimensionality is high!!

Then, the expectation may be estimated as:

$$E[X] \approx \frac{1}{N}\sum_{i=1}^{N} \hat{w}^i f(x^i) \qquad \textbf{(6)}$$

where

$$\hat{\mathrm{w}}^{\mathrm{i}} = \frac{\tilde{w}(\mathrm{x}^{\mathrm{i}})}{\frac{1}{N}\sum_{i=1}^{N}\tilde{w}(\mathrm{x}^{\mathrm{i}})} \quad \textbf{(8)} \qquad \tilde{\mathrm{w}}^{\mathrm{i}} = \frac{\tilde{p}(\mathrm{x}^{\mathrm{i}})}{\tilde{\mathrm{q}}(\mathrm{x}^{\mathrm{i}})} \quad \textbf{(7)}$$

is the **normalized importance weight** of the i_{th} particle.

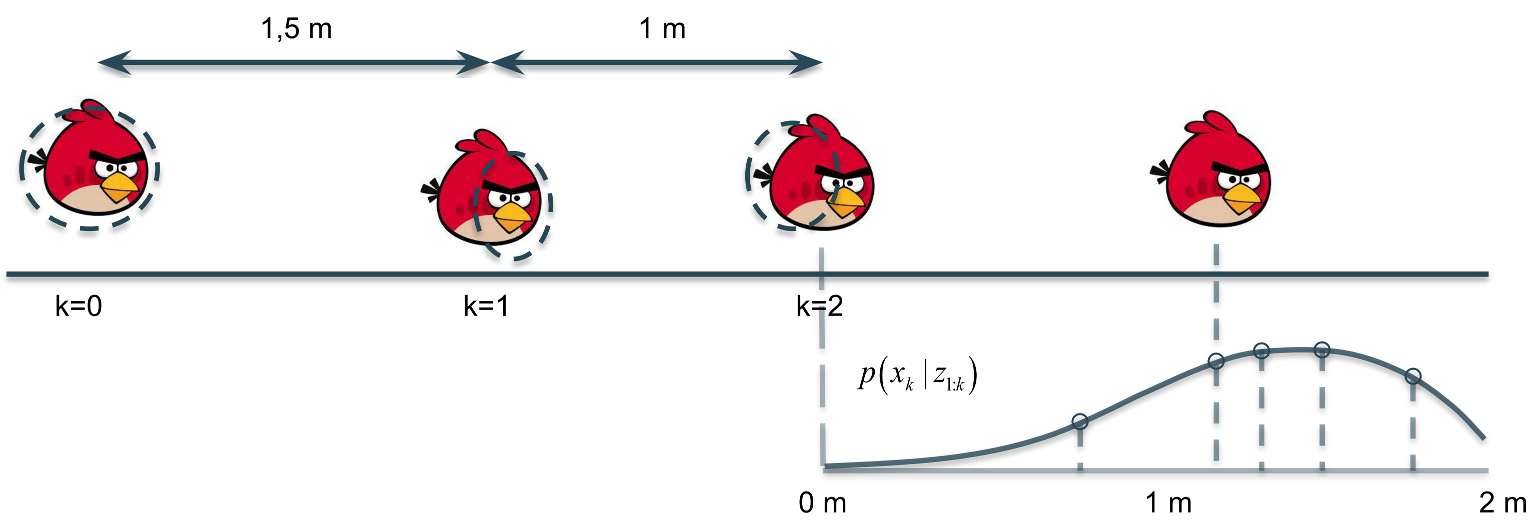
Presentation outline

- Introduction
- Monte Carlo Methods
  - Sampling
- **Motivation: Particle Filters**
  - Importance sampling
- Particle Filters
  - Generic Particle Filter
  - SIR particle filter
- Summary
- References

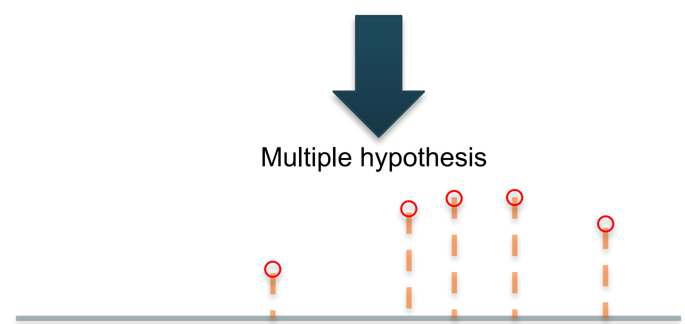
Motivation: Particle Filters

Particle Filters

- **Definition:** Particle Filters are **Sequential Monte Carlo Methods** (SMCM) that estimate the state of a system representing the posterior density function by a set of random samples (or particles) with associated weights.



$$p(x_k \mid z_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \cdot \delta\left(x_k - x_k^i\right) \qquad \textbf{(10)}$$



Importance Sampling

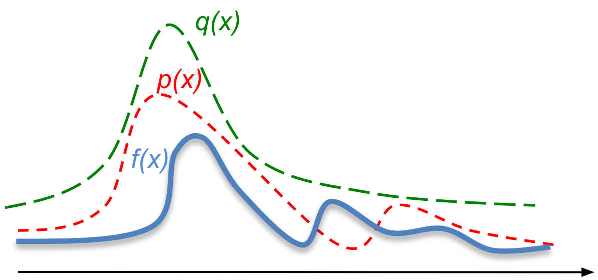
IMPORTANCE SAMPLING WITHOUT NORMALIZATION

Suppose *p(x)* is a probability density function **that is known up to a normalization factor** and from which it is difficult to draw samples.

Let *xi ~ q(x), i=1,..., Ns* be samples generated from a proposal *q(·),* **known up to a normalization factor**, called an ***importance density***.

$$p(x) = \frac{\tilde{p}(x)}{z_p} \qquad q(x) = \frac{\tilde{q}(x)}{z_q} \quad \Rightarrow \quad \int \tilde{p}(x)\,dx = z_p \qquad \int \tilde{q}(x)\,dx = z_q$$

Then, the expectation may be estimated as:



$$E[X] \approx \frac{1}{N}\sum_{i=1}^{N} \hat{w}^i f(x^i)$$

where:

$$\hat{\mathrm{w}}^{\mathrm{i}} = \frac{\tilde{w}(\mathrm{x}^{\mathrm{i}})}{\frac{1}{N}\sum_{i=1}^{N}\tilde{w}(\mathrm{x}^{\mathrm{i}})} \qquad \tilde{\mathrm{w}}^{\mathrm{i}} = \frac{\tilde{p}(\mathrm{x}^{\mathrm{i}})}{\tilde{\mathrm{q}}(\mathrm{x}^{\mathrm{i}})}$$

is the **normalized importance weight** of the ith particle.

Importance Sampling

IMPORTANCE WEIGHTS

Let us consider that samples $x_{0:k}$ were drawn from an important density $q(x_{0:k}|z_{1:k})$.
Then, weights are defined to be:

$$w^i \propto \frac{\tilde{p}(x^i)}{\tilde{q}(x^i)} \Longrightarrow w_k^i \propto \frac{\tilde{p}(x_{0:k}^i|z_{1:k})}{\tilde{q}(x_{0:k}^i|z_{1:k})} \propto \frac{p(x_{0:k}^i|z_{1:k})}{q(x_{0:k}^i|z_{1:k})}$$

These weights represent an approximation of the target distribution:

$$p\left(x_{0:k} \mid z_{1:k}\right) \approx \sum_{i=1}^{N_s} w_k^i \cdot \delta\left(x_{0:k} - x_{0:k}^i\right)$$

PROBLEM: As time increases, the amount of information that must be stored to compute these coefficients becomes intractable.

Importance Sampling

SEQUENTIAL IMPORTANCE SAMPLING

Assume that at time *k-1* we have a measurement consisting on a set of N_s weighted random samples (particles)

$$\left\{x_{1:n-1}^{(i)}\ ,\ w_{n-1}^{(i)}\right\} \quad \text{with} \quad w_{n-1}^{(i)} > 0,\ \sum_{i=1}^{N} w_{n-1}^{(i)} = 1$$

that approximate the distribution $p\left(x_{0:k-1}\right)$

The objective is to approximate $p\left(x_{0:k} \mid z_{1:k}\right)$ with a new set of samples.

$$\mathrm{w}_{\mathrm{k}}^{\mathrm{i}} \propto \frac{p(\mathrm{x}_{0:\mathrm{k}}^{\mathrm{i}} \mid \mathrm{z}_{1:\mathrm{k}})}{q(\mathrm{x}_{0:\mathrm{k}}^{\mathrm{i}} \mid \mathrm{z}_{1:\mathrm{k}})} \Rightarrow$$

Dependent of all time instants

$$\mathrm{w}_{\mathrm{k}}^{\mathrm{i}} \propto \mathrm{w}_{\mathrm{k-1}}^{\mathrm{i}}\, \Theta_{k,k-1} \Rightarrow$$

Only dependent of the current and the previous time instants

Importance Sampling

If the system can be modeled as a Markov Process, recursion allows to gather all the information of the system evolution in the importance density of the previous time instant

The **Weight Update Equation** (WUE) can be shown to be:

$$w_k^i \propto w_{k-1}^i \frac{p(z_k \mid x_k^i)\,p(x_k^i \mid x_{k-1}^i)}{q(x_k^i \mid x_{k-1}^i, z_k)} \qquad \textbf{(11)}$$

Only dependent of the current and the previous time instants. $\quad w_k^i \propto w_{k-1}^i \, \Theta_{k,k-1}$

Presentation outline

- Introduction

- Monte Carlo Methods
  - Sampling

- Motivation: Particle Filters
  - Importance sampling

- **Particle Filters**
  - Generic Particle Filter
  - SIR particle filter

- Summary

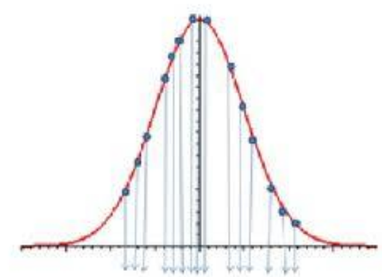
- References

Particle Filters

PARTICLE FILTERS

- The posterior density that describes the state of the system can be approximated as:

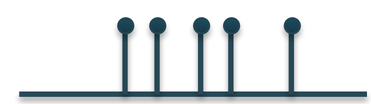
$$p(x_k \mid z_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \cdot \delta\left(x_k - x_k^i\right)$$



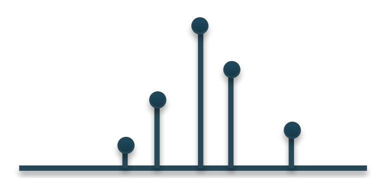
using a set of N_s particles $x_k^{\,i} \sim q(x_k \mid x_{k-1}^{\,i}, z_{1:k-1})$ with associated weights:

$$w_k^i \propto w_{k-1}^i \frac{p(z_k \mid x_k^i)\, p(x_k^i \mid x_{k-1}^i)}{q(x_k^i \mid x_{k-1}^i, z_k)} \qquad \textbf{(11)}$$

- Systematically obtain an estimation of $p(x_k \mid z_{1:k})$ in two steps:
  - **Prediction:** $p(x_k \mid z_{1:k-1}) = \int p(x_k \mid x_{k-1}, z_{1:k-1})\, dx_{k-1}$



  - **Update:** $p(x_k \mid z_{1:k}) = \dfrac{\psi(z_k \mid x_k)\, p(x_k \mid z_{1:k-1})}{p(z_k \mid z_{1:k-1})}$



Particle Filters

ESTIMATION OF THE POSTERIOR DENSITY

- **Properties**

a) As $\boldsymbol{N_s}$ **grows**, the approximation approaches the **true posterior** density $p(x_k | z_{1:k})$
b) The **dimension** of x do **not affect** the convergence
c) The variance of the weights can only increase over time

- **Problem: Degeneracy**

Repeated application over time of these two steps leads to **degeneracy** of the weights - **all the mass becomes concentrated and hence estimates are poor.**

Effective Sample Size (measure of weights variation):

$$ESS\left(\{w_k^i\}\right) = \left(\sum_{i=1}^{N} \left(w_k^i\right)^2\right)^{-1} \qquad \textbf{(12)}$$

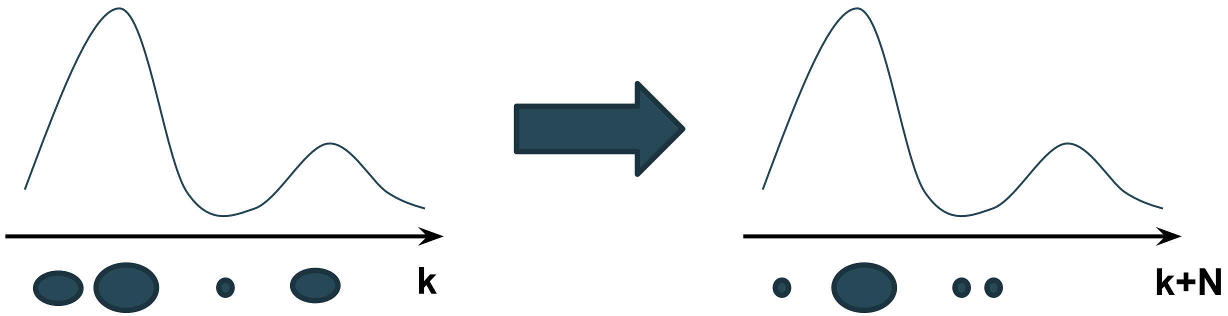
$$\underset{\text{Bad}}{\underset{\uparrow}{1}} \le ESS \le \underset{\text{Good}}{\underset{\uparrow}{N}}$$

If ESS is small (e.g. <N/2) a procedure must be applied to control the variance of the important weights: **resampling step**

Particle Filters

Degeneracy problem

• After a few iterations, all but one particle will have negligible weights.



• It can be proved that the **variance** of the importance weights can **only increase** over time unconditionally on *x*.

• It is **impossible to avoid** the degeneracy phenomenon.

• **Problems**:

- A large computational effort is devoted to update particles whose contribution to the approximation is zero.
- PDF is not correctly characterized.

Particle Filters

Degeneracy problem: Brute force

- How can we reduce the effect of degeneracy?

  a. Brute force: using a very large N_s

Often impractical!

Particle Filter representation

$$p(x_k \mid z_k) \approx \sum_{i=1}^{N_s} w_k^i \cdot \delta(x_k - x_k^i)$$

$$w_k^i \propto \frac{p(x_k \mid z_k)}{q(x_k \mid z_k)} \qquad w_k^i = \frac{1}{N_s}\frac{p(x_k \mid z_k)}{q(x_k \mid z_k)}$$

$$\lim_{Ns\to\infty}\sum_{i=1}^{N_s} w_k^i \cdot \delta(x_k - x_k^i) = \lim_{Ns\to\infty}\sum_{i=1}^{N_s}\frac{1}{N_S}\frac{p(x_k^i \mid z_k^i)}{q(x_k^i \mid z_k^i)}\delta(x_k - x_k^i) = \lim_{Ns\to\infty}\frac{p(x_k \mid z_k)}{q(x_k \mid z_k)}\sum_{i=1}^{Ns}\frac{1}{N_S}\delta(x_k - x_k^i) =$$

$$= \frac{p(x_k \mid z_k)}{q(x_k \mid z_k)}\lim_{Ns\to\infty}\left(\sum_{i=1}^{Ns}\frac{1}{N_S}\delta(x_k - x_k^i)\right) = \frac{p(x_k \mid z_k)}{q(x_k \mid z_k)}q(x_k \mid z_k) = p(x_k \mid z_k)$$

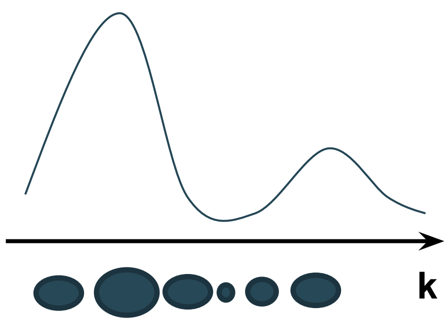
Slide credit: David Varas

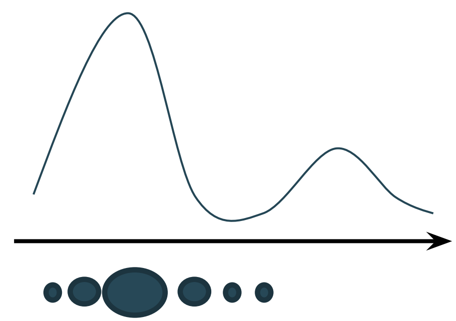
Particle Filters

Degeneracy problem

- How can we reduce the effect of degeneracy?

a. Brute force: using a very large Ns → **Often impractical!**





b. Good choice of importance density

c. Resample

Particle Filters

Degeneracy problem: good choice of importance density

- **A particular case (SIR) is to choose the importance density to be the prior:**
  - Intuitive
  - Simple to implement
  - Not optimal

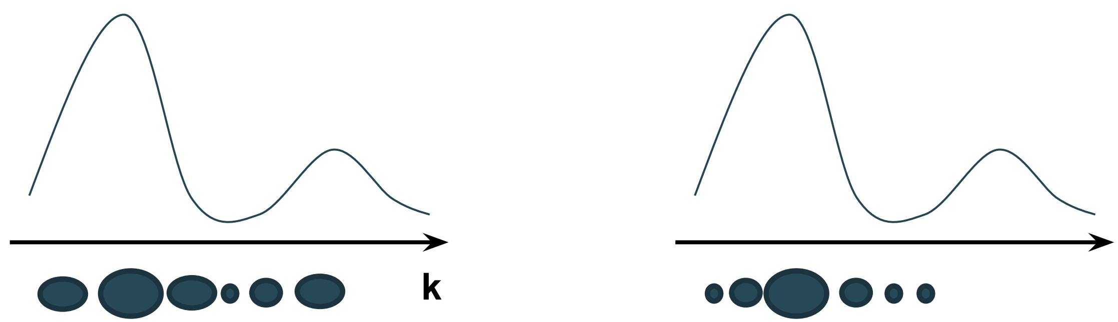
$$q(x_k \mid x_{k-1}^i, z_k) = p(x_k \mid x_{k-1}^i) \qquad \textbf{(13)}$$

Particle Filters

Degeneracy problem

- How can we reduce the effect of degeneracy?

a. Brute force: using a very large N_s → **Often impractical!**



b. Good choice of importance density →

Optimal Importance density

$$q(x_k \mid x_{k-1}^i, z_k)_{OPT} = p(x_k \mid x_{k-1}^i, z_k)$$

SIR case

$$q(x_k \mid x_{k-1}^i, z_k) = p(x_k \mid x_{k-1}^i)$$

c. Resampling

Particle Filters

Degeneracy problem: resampling

- **Basic idea**: Eliminate most particles with small weights and concentrate on particles with large weights.

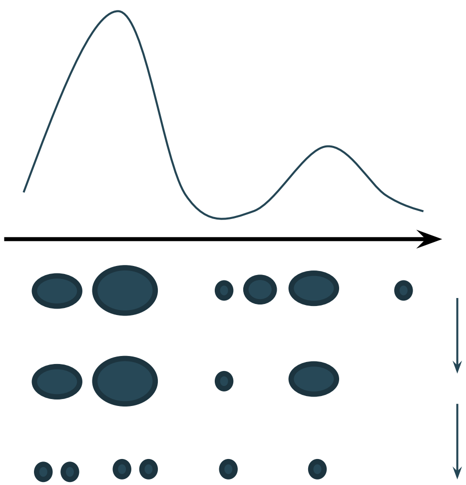
- Generate a new set $\{x_k^{i*}\}_{i=1}^{Ns}$ by **resampling with replacement** N_s times from an approximate discrete representation of $p(x_k \mid z_{1:k})$ given by

$$p(x_k \mid z_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \cdot \delta\left(x_k - x_k^i\right)$$

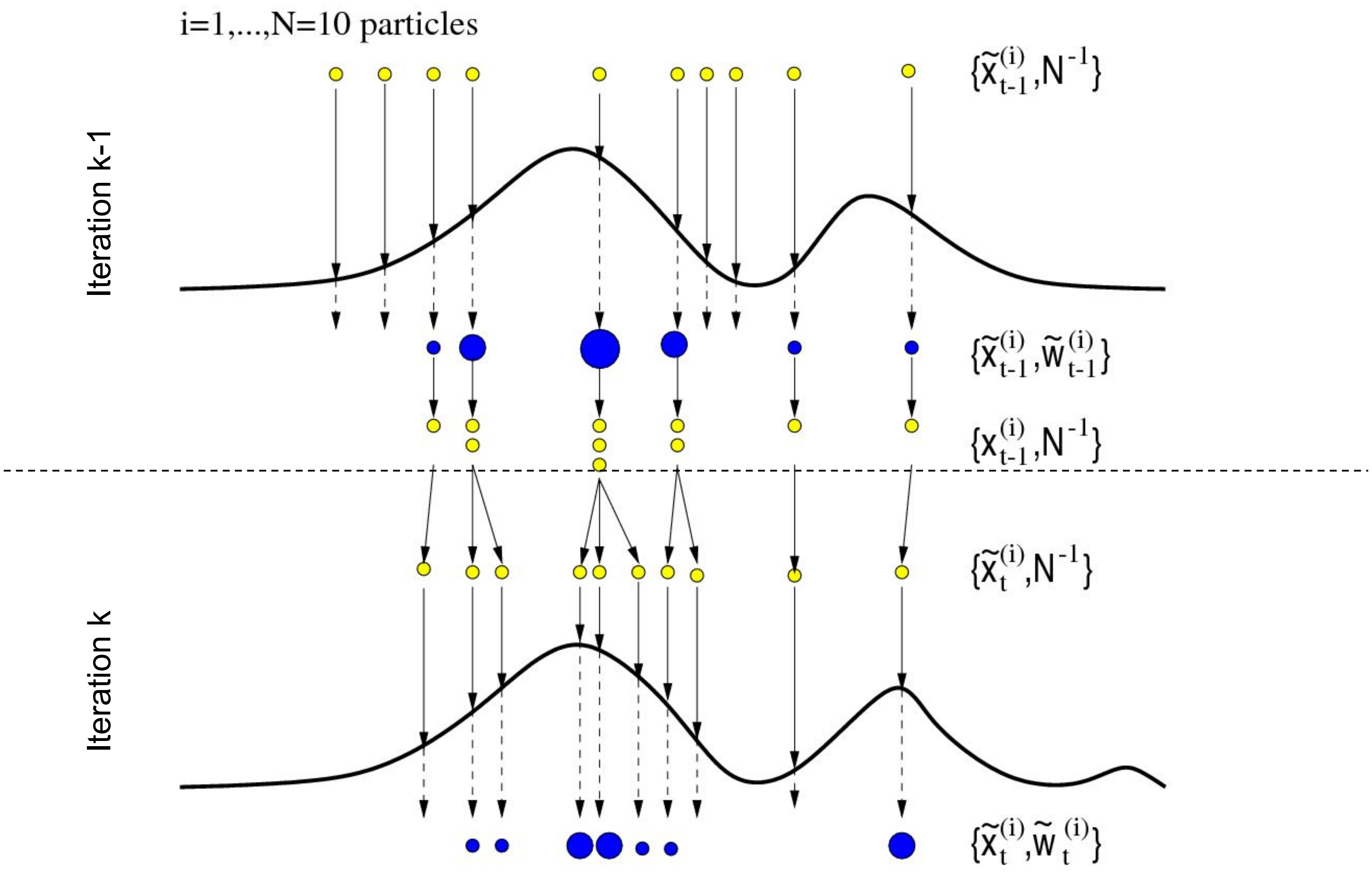
so that

$$p(x_k^{i*} = x_i^k) = w_k^i$$

i.i.d. sample from $p(x_k \mid z_{1:k})$ $\Rightarrow$ $w_k^i = \frac{1}{N_s}$



Generic Particle Filter



Particle Filters

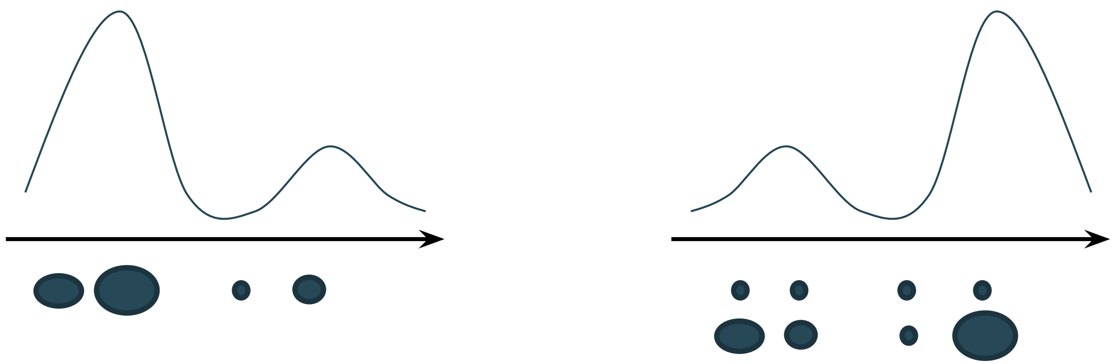
Degeneracy problem: resampling

- The contribution of the set of particles is **more efficient**

- **Eliminate** particles with **small weights** and **keep** those with **large weights**.

- Particles with **small weights can survive** to the resample step, but they will be **eliminated statistically** after a few iterations.

- It is not a problem to deal with some particles with small weights (**Diversity**).



Presentation outline

- Introduction
- Monte Carlo Methods
  - Sampling
- Motivation: Particle Filters
  - Importance sampling
- Particle Filters
  - **Generic Particle Filter**
  - SIR particle filter
- Summary
- References

Generic Particle Filter

GENERIC PARTICLE FILTER ALGORITHM

- FOR $i = 1{:}N_s$
    I. Draw $x_k^i \sim q(x_k | x_{k-1}^i, z_k)$ — Propagate
    II. Assign the particle a weight, w_k^i, according to (7) — Evaluate
- END FOR

Normalize

- Calculate total weight: $t = \text{SUM}[\{w_k^i\}_{i=1:Ns}]$
- FOR $i = 1{:}N_s$
    I. Normalize $w_k^i = t^{-1} w_k^i$
- END FOR

- Compute E[X]

Resample

- Calculate $Neff^*$ using (2)
- IF $Neff^* < N_T$
    I. Resample
- END IF

Presentation outline

- Introduction
- Monte Carlo Methods
    - Sampling
- Motivation: Particle Filters
    - Importance sampling
- Particle Filters
    - Generic Particle Filter
    - **SIR particle filter**
- Summary
- References

SIR Particle Filters

From a generic Particle Filter towards the SIR Particle Filter

- Considerations

i. Appropriate choice of **importance density**, where $q(x_k|x^i_{k-1},z_k)$ is chosen to be $p(x_k|x^i_{k-1})$

$$q(x_k|x^i_{k-1},z_k) = p(x_k|x^i_{k-1})$$

ii. **Resampling** step applied at every time index

$$w^i_{k-1} = \frac{1}{N_s} \quad \forall i$$

Implications of (i) and (ii) in the **Weight Update Equation**:

$$w_k^i \propto w_{k-1}^i \frac{p(z_k \mid x_k^i)\, p(x_k^i \mid x_{k-1}^i)}{q(x_k^i \mid x_{k-1}^i, z_k)} \overset{\text{(i)}}{\propto} w_{k-1}^i \frac{p(z_k \mid x_k^i)\, p(x_k^i \mid x_{k-1}^i)}{p(x_k^i \mid x_{k-1}^i)} \propto w_{k-1}^i\, p(z_k \mid x_k^i)$$

$$w_k^i \propto w_{k-1}^i\, p(z_k \mid x_k^i) \overset{\text{(ii)}}{\propto} \frac{1}{N_s}\, p(z_k \mid x_k^i) \propto p(z_k \mid x_k^i) \implies \boxed{w_k^i \propto p(z_k \mid x_k^i)} \quad \textbf{(14)}$$

SIR Particle Filters

SIR Particle Filter

- Algorithm

▪FOR $i = 1{:}N_s$

I. Draw $x_k^i \sim p(x_k|x^i_{k-1})$ — Propagate

II. Calculate $w^i_k = p(z_k|x^i_k)$ — Evaluate

▪END FOR

▪Calculate total weight: $t = \text{SUM}[\{w^i_k\}_{i=1:Ns}]$

▪FOR $i = 1{:}N_s$ — Normalize

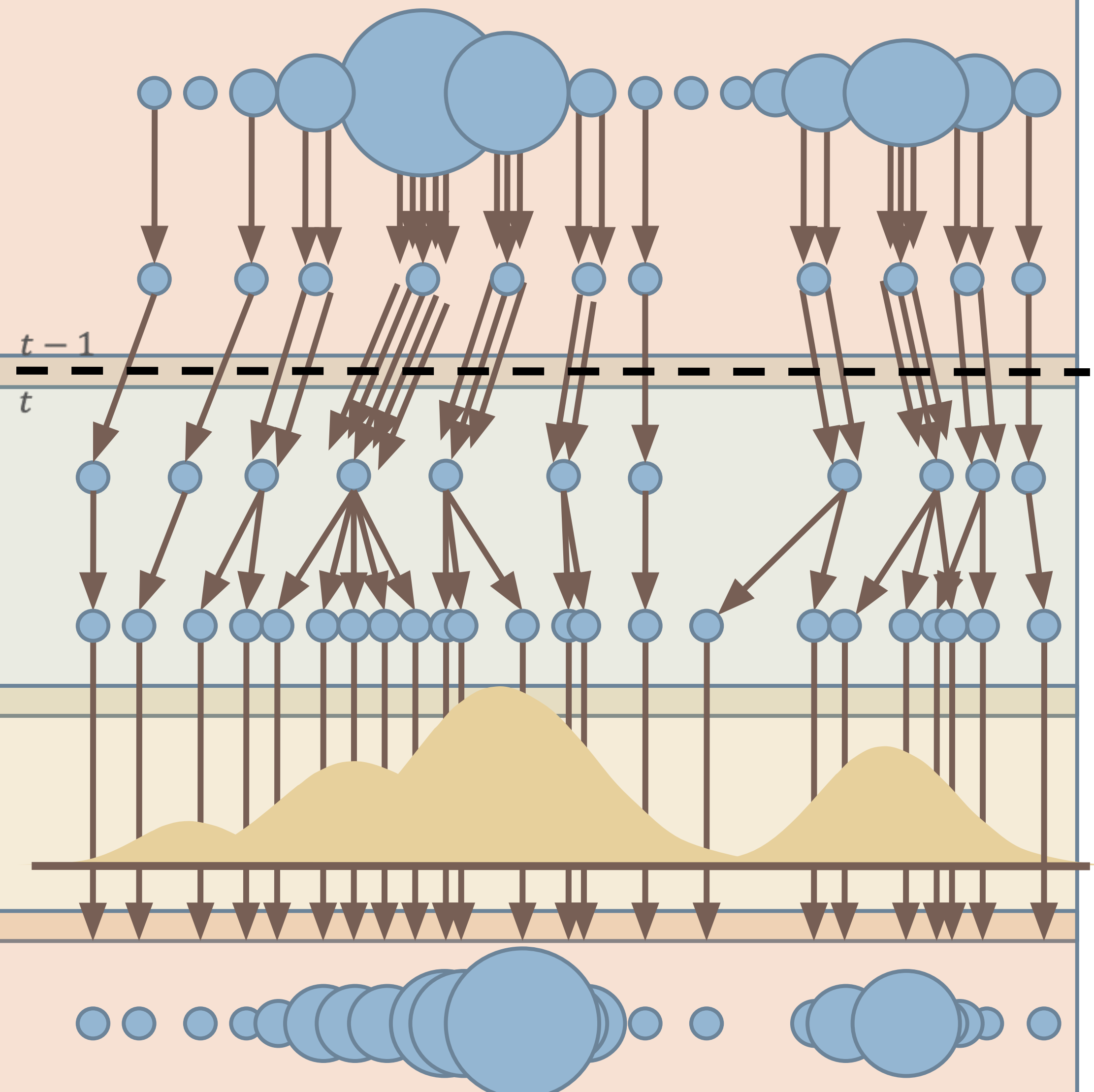
I. Normalize $w^i_k = t^{-1} w^i_k$

▪END FOR

▪Resample — Resample

Generic Particle Filter

- **Begin** with weighted samples from t-1
- **Resample:** draw samples according to $\{w_{t-1}\}^{n=1:N}$
- **Drift:** apply motion model (no noise)
- **Diffuse:** apply noise to spread particles
- **Measure:** weights are assigned by likelihood response & normalized
- **Finish:** density estimate



SIR Particle Filters

SIR Particle Filter

- Characteristics
  - Importance density independent of measurement z_k

State space explored without any knowledge of the observations

a. Inefficient
b. Sensitive to outliers

  - Resampling step applied at each iteration

a. Loss of particles diversity

  - Direct access to $p(z_k|x_k)$

a. Efficient computation of w_k^i
b. Importance density easily sampled

Particle Filters

D. Klein, D. Schulz, S. Frintrop, and A. Cremers, Adaptive Real-Time Video Tracking for Arbitrary Objects , International Conference on Intelligent Robots and Systems (IROS), 2010

Presentation outline

- Introduction
- Monte Carlo Methods
    - Sampling
- Motivation: Particle Filters
    - Importance sampling
- Particle Filters
- Generic Particle Filter
- SIR particle filter
- **Summary**
- References

Summary

- **Monte Carlo Methods** provide algorithms to estimate complex pdf.

- **Particle Filters** are **Sequential Monte Carlo Methods** for estimating the state of a system with the function $p(x_k| z_{1:k})$.

- As N_s grows, the approximation **approaches the true posterior density** $p(x_k| z_{1:k})$

- **No assumptions** are made about the model and the noise.

- The **dimension** of x do not affect in the **convergence**.

- After a few iterations, the **degeneracy problem** appears (**Resampling step**).

- Particle Filters are **rapid** and **robust** algorithms able to **estimate complex pdf**.

References

- A.Doucet, N. de Freitas and N.Gordon (Editors),
  Sequential Monte Carlo methods in practice
  Springer, 2001
- P.M.Djuric, J.H.Kotecha, J.Zhang, Y.Huang, T.Ghirmai, M.F.Bugallo, J.Miguez
  "**Particle filtering**"
  IEEE Signal Processing Magazine, September 2003
- A.Doucet and X.Wang,
  "**Monte Carlo Methods for Signal Processing**"
  IEEE Signal Processing Magazine, November 2005
- X.Wang, R.Chen, J.S.Liu
  "**Monte Carlo Bayesian signal processing for wireless communications**"
  Journal of VLSI Signal Processing, Vol.30, pp.89-105, 2002
  Kluwer Academic Publishers
- S. Arulampalam, S. Maskell, N. Gordon, T. Clapp
  "**A tutorial on Particle Filters for on-line non-linear/non-Gaussian Bayesian tracking**"
  IEEE Trans. On Signal Processing, Feb. 2002.