A brief Introduction to Particle Filters

Michael Pfeiffer
pfeiffer@igi.tugraz.at

18.05.2004
Agenda

• Problem Statement
• Classical Approaches
• Particle Filters
  – Theory
  – Algorithms
• Applications
Problem Statement

• Tracking the state of a system as it evolves over time

• We have: Sequentially arriving (noisy or ambiguous) observations

• We want to know: Best possible estimate of the hidden variables
Illustrative Example: Robot Localization

Sensory model: never more than 1 mistake
Motion model: may not execute action with small prob.
Illustrative Example: Robot Localization

$t=1$
Illustrative Example: Robot Localization

t=2
Illustrative Example: Robot Localization

t=3
Illustrative Example: Robot Localization

$t=4$
Illustrative Example: Robot Localization

Trajectory

Prob

0 1

t=5
Applications

• Tracking of aircraft positions from radar
• Estimating communications signals from noisy measurements
• Predicting economical data
• Tracking of people or cars in surveillance videos
Bayesian Filtering / Tracking Problem

- Unknown State Vector $x_{0:t} = (x_0, \ldots, x_t)$
- Observation Vector $z_{1:t}$
- Find PDF $p(x_{0:t} | z_{1:t})$ ... *posterior distribution*
- or $p(x_t | z_{1:t})$ ... *filtering distribution*

- Prior Information given:
  - $p(x_0)$ ... prior on state distribution
  - $p(z_t | x_t)$ ... sensor model
  - $p(z_t | x_{t-1})$ ... Markovian state-space model
Sequential Update

• Storing all incoming measurements is inconvenient

• Recursive filtering:
  - **Predict** next state pdf from current estimate
  - **Update** the prediction using sequentially arriving new measurements

• **Optimal** Bayesian solution: recursively calculating exact posterior density
Bayesian Update and Prediction

• Prediction

\[ p(x_t \mid z_{1:t-1}) = \int p(x_t \mid x_{t-1}) p(x_{t-1} \mid z_{1:t-1}) \, dx_{t-1} \]

• Update

\[ p(x_t \mid z_{1:t}) = \frac{p(z_t \mid x_t) p(x_t \mid z_{1:t-1})}{p(z_t \mid z_{1:t-1})} \]

\[ p(z_t \mid z_{1:t-1}) = \int p(z_t \mid x_t) p(x_t \mid z_{1:t-1}) \, dx_t \]
Agenda

• Problem Statement
• **Classical Approaches**
• Particle Filters
  – Theory
  – Algorithms
• Applications
Kalman Filter

- Optimal solution for linear-Gaussian case

- Assumptions:
  - State model is known \textit{linear} function of last state and \textit{Gaussian noise} signal
  - Sensory model is known \textit{linear} function of state and \textit{Gaussian noise} signal
  - Posterior density is \textit{Gaussian}
Kalman Filter: Update Equations

\[
x_t = F_t x_{t-1} + v_{t-1} \quad v_{t-1} \sim N(0, Q_{t-1})
\]

\[
z_t = H_t x_t + n_t \quad n_t \sim N(0, R_t)
\]

\(F_t, H_t\) : known matrices

\[
p(x_{t-1} \mid z_{1:t-1}) = N(x_{t-1} \mid m_{t-1|t-1}, P_{t-1|t-1})
\]

\[
p(x_t \mid z_{1:t-1}) = N(x_t \mid m_{t-1|t-1}, P_{t|t-1})
\]

\[
p(x_t \mid z_{1:t}) = N(x_t \mid m_{t|t}, P_{t|t})
\]

\[
m_{t|t-1} = F_t m_{t-1|t-1}
\]

\[
P_{t|t-1} = Q_{t-1} + F_t P_{t-1|t-1} F_t^T
\]

\[
m_{t|t} = m_{t|t-1} + K_t (z_t - H_t m_{t|t-1})
\]

\[
P_{t|t} = P_{t|t-1} - K_t H_t P_{t|t-1}
\]

\[
S_t = H_t P_{t|t-1} H_t^T + R_t
\]

\[
K_t = P_{t|t-1} H_t^T S_t^{-1}
\]
Limitations of Kalman Filtering

- Assumptions are too strong. We often find:
  - Non-linear Models
  - Non-Gaussian Noise or Posterior
  - Multi-modal Distributions
  - Skewed distributions

- Extended Kalman Filter:
  - Local linearization of non-linear models
  - Still limited to Gaussian posterior
Grid-based Methods

- Optimal for discrete and finite state space
- Keep and update an estimate of posterior pdf for every single state
- No constraints on posterior (discrete) density
Limitations of Grid-based Methods

- Computationally expensive
- Only for finite state sets
- Approximate Grid-based Filter
  - divide continuous state space into finite number of cells
  - Hidden Markov Model Filter
  - Dimensionality increases computational costs dramatically
Agenda

• Problem Statement
• Classical Approaches
• **Particle Filters**
  - Theory
  - Algorithms
• Applications
Many different names…

Particle Filters

• (Sequential) Monte Carlo filters
• Bootstrap filters
• Condensation
• Interacting Particle Approximations
• Survival of the fittest
• …
Sample-based PDF Representation

- **Monte Carlo** characterization of pdf:
  - Represent posterior density by a set of random i.i.d. samples (particles) from the pdf $p(x_{0:t} | z_{1:t})$
  - For larger number $N$ of particles equivalent to functional description of pdf
  - For $N \to \infty$ approaches optimal Bayesian estimate
Sample-based PDF Representation

- Regions of high density
  - Many particles
  - Large weight of particles
- Uneven partitioning
- Discrete approximation for continuous pdf

\[
P_N(x_{0:t} \mid z_{1:t}) = \sum_{i=1}^{N} w_i \delta(x_{0:t} - x_{0:t}^i)
\]
Importance Sampling

- Draw $N$ samples $x_{0:t}^{(i)}$ from Importance sampling distribution $\pi(x_{0:t} | z_{1:t})$

- Importance weight

- Estimation of arbitrary functions $f_t$:

\[
\hat{I}_N(f_t) = \sum_{i=1}^{N} f_t(x_{0:t}^{(i)}) \tilde{w}_t^{(i)}, \quad \tilde{w}_t^{(i)} = \frac{w(x_{0:t}^{(i)})}{\sum_{j=1}^{N} w(x_{0:t}^{(j)})}
\]

\[
\hat{I}_N(f_t) \xrightarrow{a.s.} I(f_t) = \int f_t(x_{0:t}) p(x_{0:t} | y_{1:t}) dx_{0:t}
\]
Sequential Importance Sampling (SIS)

- **Augmenting the samples**

\[
\pi(x_{0:t} \mid z_{1:t}) = \pi(x_{0:t-1} \mid z_{1:t-1}) \pi(x_t \mid x_{0:t-1}, z_{1:t}) = \\
\pi(x_{0:t-1} \mid z_{1:t-1}) \pi(x_t \mid x_{t-1}, z_t)
\]

\(x_t^{(i)} \sim \pi(x_t \mid x_{t-1}^{(i)}, z_t)\)

- **Weight update**

\[
w_t^{(i)} \propto w_{t-1}^{(i)} \frac{p(z_t \mid x_t^{(i)}) p(x_t^{(i)} \mid x_{t-1}^{(i)})}{\pi(x_t^{(i)} \mid x_{t-1}^{(i)}, z_t)}
\]
Degeneracy Problem

- After a few iterations, all but one particle will have negligible weight.
- Measure for degeneracy: *Effective sample size*

\[
N_{\text{eff}} = \frac{N}{1 + \text{Var}(w_t^*)}
\]

- Small \( N_{\text{eff}} \) indicates severe degeneracy.
- Brute force solution: Use very large \( N \).
Choosing Importance Density

• Choose $\pi$ to minimize variance of weights

• Optimal solution: $\pi_{opt}(x_t | x_{t-1}^{(i)}, z_t) = p(x_t | x_{t-1}^{(i)}, z_t)$
  \[ \Rightarrow w_t^{(i)} \propto w_{t-1}^{(i)} p(z_t | x_{t-1}^{(i)}) \]

• Practical solution: $\pi(x_t | x_{t-1}^{(i)}, z_t) = p(x_t | x_{t-1}^{(i)})$
  \[ \Rightarrow w_t^{(i)} \propto w_{t-1}^{(i)} p(z_t | x_t^{(i)}) \]

  - importance density = prior
Resampling

- Eliminate particles with small importance weights
- Concentrate on particles with large weights

- Sample N times with replacement from the set of particles $x_t^{(i)}$ according to importance weights $w_t^{(i)}$

- "Survival of the fittest"
Sampling Importance Resample Filter: Basic Algorithm

1. INIT, t=0
   - for i=1,..., N: sample $x_0^{(i)} \sim p(x_0)$; t:=1;

2. IMPORTANCE SAMPLING
   - for i=1,..., N: sample $x_t^{(i)} \sim p(x_t|x_{t-1}^{(i)})$
     - $x_{0:t}^{(i)} := (x_{0:t-1}^{(i)}, x_t^{(i)})$
   - for i=1,..., N: evaluate importance weights $w_t^{(i)}=p(z_t|x_t^{(i)})$
     - Normalize the importance weights

3. SELECTION / RESAMPLING
   - resample with replacement N particles $x_{0:t}^{(i)}$ according to the importance weights
   - Set t:=t+1 and go to step 2
Variations

- **Auxiliary Particle Filter:**
  - resample at time t-1 with one-step lookahead (re-evaluate with new sensory information)

- **Regularisation:**
  - resample from continuous approximation of posterior $p(x_t | z_{1:t})$
Visualization of Particle Filter

- Unweighted measure
- Compute importance weights $\Rightarrow p(x_{t-1}|z_{1:t-1})$
- Resampling
- Move particles
- Predict $p(x_t|z_{1:t-1})$
Particle Filter Demo 1

moving Gaussian + uniform, N=100 particles
Particle Filter Demo 2

moving Gaussian + uniform, N=1000 particles
Particle Filter Demo 3

moving (sharp) Gaussian + uniform, N=100 particles
Particle Filter Demo 4

moving (sharp) Gaussian + uniform, N=1000 particles
Particle Filter Demo 5

mixture of two Gaussians,
filter loses track of smaller and less pronounced peaks
Obtaining state estimates from particles

- Any estimate of a function \( f(x_t) \) can be calculated by discrete PDF-approximation

\[
E[f(x_t)] = \frac{1}{N} \sum_{j=1}^{N} w_t^{(j)} f(x_t^{(j)})
\]

- Mean:

\[
E[x_t] = \frac{1}{N} \sum_{j=1}^{N} w_t^{(j)} x_t^{(j)}
\]

- MAP-estimate: particle with largest weight

- Robust mean: mean within window around MAP-estimate
Pros and Cons of Particle Filters

+ Estimation of full PDFs
+ Non-Gaussian distributions
  + e.g. multi-modal
+ Non-linear state and observation model
+ Parallelizable

- Degeneracy problem
- High number of particles needed
- Computationally expensive
- Linear-Gaussian assumption is often sufficient
Agenda

• Problem Statement
• Classical Approaches
• Particle Filters
  – Theory
  – Algorithms
• Applications
Mobile Robot Localization

- Animation by Sebastian Thrun, Stanford
- [http://robots.stanford.edu](http://robots.stanford.edu)
Positioning Systems

• Track car position in given road map
• Track car position from radio frequency measurements
• Track aircraft position from estimated terrain elevation
• Collision Avoidance (Prediction)
• Replacement for GPS

Model Estimation

• Tracking with multiple motion-models
  - Discrete hidden variable indicates active model (manoeuvre)

• Recovery of signal from noisy measurements
  - even if signal may be absent (e.g. synaptic currents)
  - mixture model of several hypotheses

• Neural Network model selection [de Freitas]¹
  - estimate parameters and architecture of RBF network from input-output pairs
  - on-line classification (time-varying classes)

Other Applications

- Visual Tracking
  - e.g. human motion (body parts)
- Prediction of (financial) time series
  - e.g. mapping gold price → stock price
- Quality control in semiconductor industry
- Military applications
  - Target recognition from single or multiple images
  - Guidance of missiles
Possible Uses for our Group

- Reinforcement Learning
  - POMDPs
  - Estimating Opponent States
- RoboCup: Multi-robot localization and tracking
- Representation of PDFs
- Prediction Tasks
- Preprocessing of Visual Input
- Identifying Neural Network Layouts or other Hidden System Parameters
- Applications in Computational Neuroscience (?)

- Other suggestions?
Sources


Thank you!