Quasi-Monte Carlo particle filters on multi-core architectures

Second SHARCNET Symposium on GPU and Cell computing

R. G. Esteves

University of Waterloo, RapidMind Corporation

May 20th, 2009

Outline

Meaning from signals

Signal	Semantics
time-indexed values	handwritten text/math financial significance medical situation
location,intensity pairs time indexed images	identification of medical structures identification of dangerous objects behaviour recognition

Describing signals

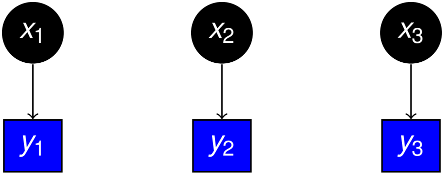
- Measurements are uncertain
- Great variability
  - spatial differences - scale
  - temporal differences - speed, duration
  - context difference - background

Model-based pattern recognition

- Instead of analyzing observed signal
- Posit (latent) process ("model") that generates signal

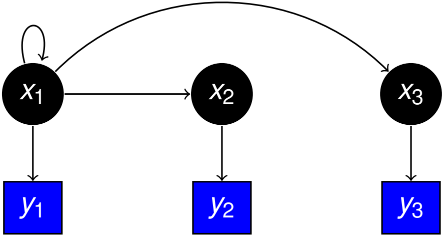
Model-based pattern recognition

- Instead of analyzing observed signal
- Posit (latent) process ("model") that generates signal



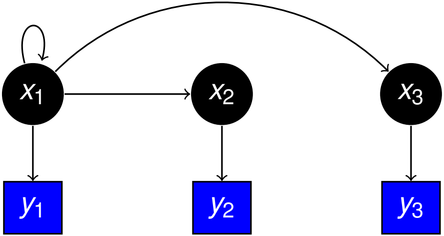
Model-based pattern recognition

- Instead of analyzing observed signal
- Posit (latent) process ("model") that generates signal



Model-based pattern recognition

- Instead of analyzing observed signal
- Posit (latent) process ("model") that generates signal



Transition $p(x_t|x_{t-1})$

Observation $p(y_t|x_t)$

The estimation problem

$$
\begin{aligned}
p(x_t|y_{0:t}) &\propto p(y_t|x_t)p(x_t|y_{0:t-1}) \\
p(x_t|y_{0:t-1}) &= \int p(x_t|x_{t-1})p(x_{t-1}|y_{0:t-1})\mathrm{d}x_{t-1}
\end{aligned}
$$

The estimation problem

$$
\begin{aligned}
p(x_t|y_{0:t}) &\propto p(y_t|x_t)p(x_t|y_{0:t-1}) \\
p(x_t|y_{0:t-1}) &= \int p(x_t|x_{t-1})p(x_{t-1}|y_{0:t-1})\mathrm{d}x_{t-1}
\end{aligned}
$$

- If transition/observation models are linear/Gaussian, solve by Kalman filter
- KF allows optimal solution
- Common approach for nonlinear/non-Gaussian problems: linearize, KF
- Nonlinear case handled directly by SMC methods

The estimation problem

$$
\begin{aligned}
p(x_t|y_{0:t}) &\propto p(y_t|x_t)p(x_t|y_{0:t-1}) \\
p(x_t|y_{0:t-1}) &= \int p(x_t|x_{t-1})p(x_{t-1}|y_{0:t-1})\mathrm{d}x_{t-1}
\end{aligned}
$$

- If transition/observation models are linear/Gaussian, solve by Kalman filter
- KF allows optimal solution
- Common approach for nonlinear/non-Gaussian problems: linearize, KF
- Nonlinear case handled directly by SMC methods

Particle filter overview

- Sequential Monte Carlo (SMC, or Particle Filter)
- Enable approximate solution to non-linear systems
- Better to have approximate solution to real problem than exact solution to approximate problem
- Maintain discrete, or *particle* representation of filtering distribution
- Founded on Bayesian principles
  - Domain-knowledge and other prior information readily incorporated
  - Results are full distributions, not point- or interval-estimates
- Sampling techniques are (significantly) more computationally demanding than analytical solutions.

Particle filter overview

- Sequential Monte Carlo (SMC, or Particle Filter)
- Enable approximate solution to non-linear systems
- Better to have approximate solution to real problem than exact solution to approximate problem
- Maintain discrete, or *particle* representation of filtering distribution
- Founded on Bayesian principles
  - Domain-knowledge and other prior information readily incorporated
  - Results are full distributions, not point- or interval-estimates
- Sampling techniques are (significantly) more computationally demanding than analytical solutions.

Particle filter overview

- Sequential Monte Carlo (SMC, or Particle Filter)
- Enable approximate solution to non-linear systems
- Better to have approximate solution to real problem than exact solution to approximate problem
- Maintain discrete, or *particle* representation of filtering distribution
- Founded on Bayesian principles
  - Domain-knowledge and other prior information readily incorporated
  - Results are full distributions, not point- or interval-estimates
- Sampling techniques are (significantly) more computationally demanding than analytical solutions.

Particle filter overview

- Sequential Monte Carlo (SMC, or Particle Filter)
- Enable approximate solution to non-linear systems
- Better to have approximate solution to real problem than exact solution to approximate problem
- Maintain discrete, or *particle* representation of filtering distribution
- Founded on Bayesian principles
  - Domain-knowledge and other prior information readily incorporated
  - Results are full distributions, not point- or interval-estimates
- Sampling techniques are (significantly) more computationally demanding than analytical solutions.

Particle filter overview

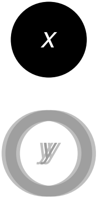
- Sequential Monte Carlo (SMC, or Particle Filter)
- Enable approximate solution to non-linear systems
- Better to have approximate solution to real problem than exact solution to approximate problem
- Maintain discrete, or *particle* representation of filtering distribution
- Founded on Bayesian principles
  - Domain-knowledge and other prior information readily incorporated
  - Results are full distributions, not point- or interval-estimates
- Sampling techniques are (significantly) more computationally demanding than analytical solutions.

Particle filter overview

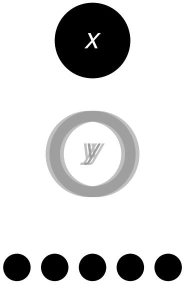
- Sequential Monte Carlo (SMC, or Particle Filter)
- Enable approximate solution to non-linear systems
- Better to have approximate solution to real problem than exact solution to approximate problem
- Maintain discrete, or *particle* representation of filtering distribution
- Founded on Bayesian principles
  - Domain-knowledge and other prior information readily incorporated
  - Results are full distributions, not point- or interval-estimates
- Sampling techniques are (significantly) more computationally demanding than analytical solutions.

Particle filters

Particle filters

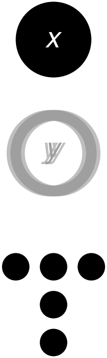


Particle filters



Particle filters

Particle filters



Particle filter algorithm

for $t \in \{2, \ldots T\}$ **do**

  Sample N particles $\{\tilde{x}_t^{(i)}\}_{i=1}^N \sim p(x_t | x_{t-1})$

  Likelihood evaluation $\omega_t^{(i)} = \frac{p(y_t|\tilde{x}_t^{(i)})}{\sum_{i=1}^N p(y_t|\tilde{x}_t^{(i)})}$

  $\{x_t^{(i)}\}_{i=1}^N \leftarrow$ RESAMPLING($\tilde{X}$, $\{\omega^{(i)}\}$)

end for

function RESAMPLING(X,Ω)

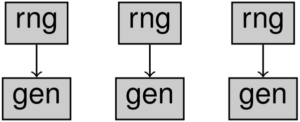
  index $\sim$ Multinomial(Ω)

  return DIFFUSE($\{x^{(i)} | x^{(i)} \in X \bigwedge i \in \text{index}\}$)

end function

Parallel particle filters

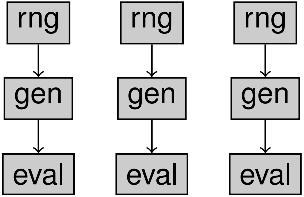
Particle dynamics



Parallel particle filters

Particle dynamics

Likelihood evaluation

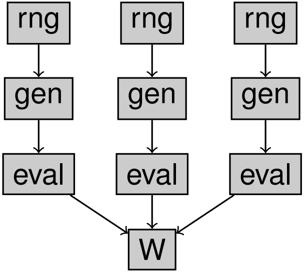


Parallel particle filters

Particle dynamics

Likelihood evaluation

Resampling

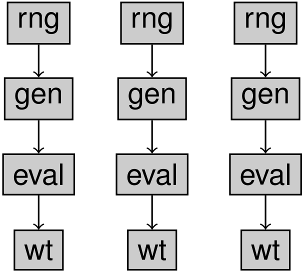


Parallel particle filters

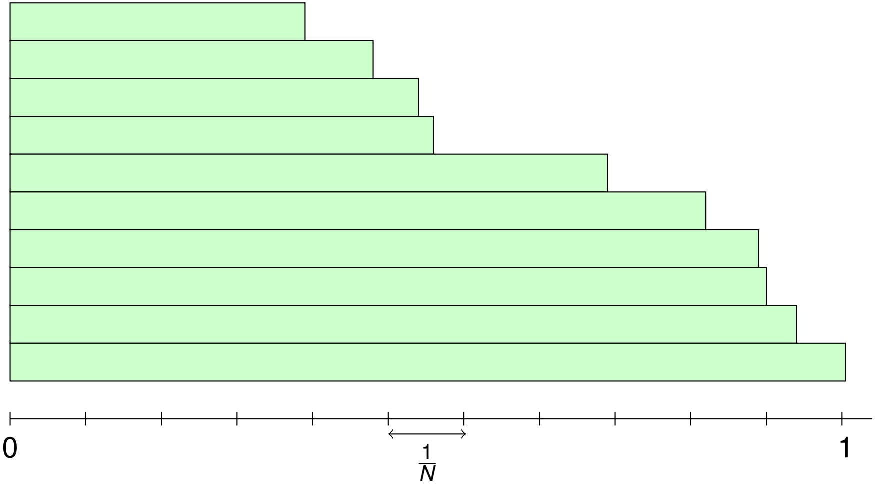
Particle dynamics

Likelihood evaluation

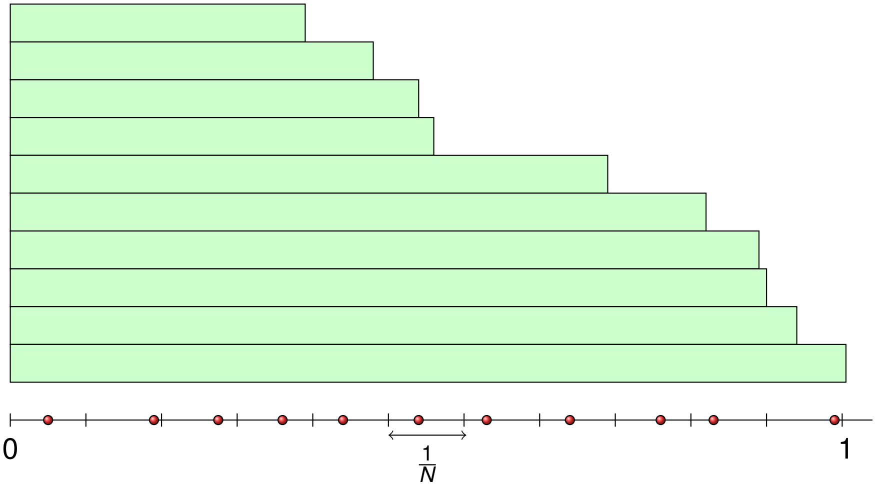
Resampling



Parallel resampling methods



Parallel resampling methods



Are parallel algorithms enough to cope with problem size?

- Inference problems scale exponentially ("curse of dimensionality")
- Parallel machines only offer solutions that scale linearly
- Additional improvements are needed

Are parallel algorithms enough to cope with problem size?

- Inference problems scale exponentially ("curse of dimensionality")
- Parallel machines only offer solutions that scale linearly
- Additional improvements are needed

Increasing efficiency of PF

- Reducing time to process each particle
  - Subspace sampling
  - Simplify likelihood ratio evaluation
- Reduce number of particles necessary to achieve accuracy
  - Improve guesses – importance sampling
  - Application-specific heuristics (local search)
  - Variance reduction

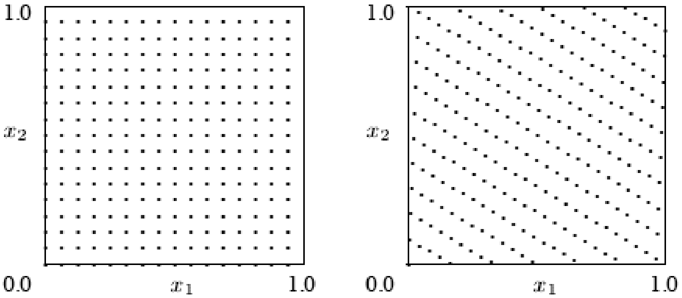
Increasing efficiency of PF

- Reducing time to process each particle
  - Subspace sampling
  - Simplify likelihood ratio evaluation
- Reduce number of particles necessary to achieve accuracy
  - Improve guesses – importance sampling
  - Application-specific heuristics (local search)
  - Variance reduction

Quasi-Monte Carlo techniques

- Variance reduction technique
- Lattice particle filters [Lemieux et al., 2001]
- Sequence driving simulation is now not independent, (randomized) highly-uniform
- Can use components of state that result in most significant variation ("effective dimension")
- Fight "over-adaptation" of traditional SIR PF.

Low-discrepancy sequences



Results

Tracker: 4D state, (x, y, w, h)

Implementation	N	Error	Performance
Serial SIR	10^3	3.8	1
Parallel SIR (G94)	10^3	3.3	15.3
Parallel LPF (G94)	10^3	0.7	9.2

Future work

- Experiment with more localized resampling algorithms
- Trajectory recognition using multi-core host/GPU array
- Sensitivity analysis for better choice of lattice rule

Conclusions

- Bayesian signal processing is a very powerful and general tool
- Sequential Monte Carlo methods allow for flexible Bayesian inference
- Parallel processors are crucial to enabling inferencing
- Parallel processing alone cannot handle curse of dimensionality