8.6 Particle Filters

048825: Estimation and Identification in Dynamic Systems
Nahum Shimkin, Spring 2016

Outline

- The nonlinear estimation problem
- Monte-Carlo Sampling:
  - Crude Monte-Carlo
  - Importance Sampling
- Sequential Monte-Carlo

The Estimation Problem:

Consider a discrete-time stochastic system with state $\mathbf{x}_t \in \mathbb{R}^n$, and observation $\mathbf{y}_t \in \mathbb{R}^m$. The system is specified by the following prior data:

$p(\mathbf{x}_o)$: initial state distribution

$p(\mathbf{x}_{t+1} | \mathbf{x}_t)$: state transition law $(t=0, 1, 2, ..)$

$p(\mathbf{y}_t | \mathbf{x}_t)$: measurement distribution $(t=1, 2, ..)$

Denote

$$\mathbf{x}_{0:t} = \{\mathbf{x}_0, \ldots, \mathbf{x}_t\}, \quad \mathbf{y}_{1:t} = \{\mathbf{y}_1, \ldots, \mathbf{y}_t\}.$$

3

The Estimation Problem (cont'd)

Note:

- As usual, we assume the Markovian state property:

  $$p(\mathbf{x}_{t+1} | \mathbf{x}_{0:t}, \mathbf{y}_{1:t}) = p(\mathbf{x}_{t+1} | \mathbf{x}_t), \quad p(\mathbf{y}_t | \mathbf{x}_{0:t}, \mathbf{y}_{1:t-1}) = p(\mathbf{y}_t | \mathbf{x}_t)$$

- The above laws are often specified through state equations: $\mathbf{x}_{t+1} = f(\mathbf{x}_t, w_t); \quad \mathbf{y}_t = h(\mathbf{x}_t, v_t)$
- The system may be time-varying, for simplicity we omit the time variable.

4

The Estimation Problem (cont'd)

- We wish to estimate the state $\mathbf{x}_t$, based on the prior data and the observations $\mathbf{y}_{1:t}$.
- The MMSE-optimal estimator, for example, is given by $\hat{\mathbf{x}}_t = E(\mathbf{x}_t \mid \mathbf{y}_{1:t})$ with covariance $P_t = \mathrm{cov}(\mathbf{x}_t \mid \mathbf{y}_{1:t})$.
- More generally, compute the posterior state distribution: $\hat{p}_t(\mathbf{x}_t) = p(\mathbf{x}_t \mid \mathbf{y}_{1:t}), \quad \mathbf{x}_t \in \mathbf{X}$

We can then compute

$$\hat{\mathbf{x}}_t = \int_{\mathbf{X}} \mathbf{x}\, \hat{p}_t(\mathbf{x})\, dx \quad \text{(and its covariance)}$$

$$\hat{\mathbf{x}}_t^{\mathrm{MAP}} = \arg\max_x \hat{p}(\mathbf{x}_t), \quad \text{(etc.)}$$

5

Posterior Distribution Formulas:

Using the Bayes rule, we easily obtain the following two-step recursive formulas that relate $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ to $p(\mathbf{x}_{t-1}|\,\mathbf{y}_{1:t-1})$ and $\mathbf{y}_t$:

- Time Update (prediction step):

$$p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1}) = \int_{x_t} p(\mathbf{x}_t \mid \mathbf{x}_{t-1})\, p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1})\, d\mathbf{x}_{t-1}$$

- Measurement Update:

$$p(\mathbf{x}_t \mid \mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t \mid \mathbf{x}_t) p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})}{\int_{\mathbf{x}_1} p(\mathbf{x}_t \mid \mathbf{x}_t) p(\mathbf{x}_t \mid \mathbf{x}_{1:t-1})} d\mathbf{x}_t$$

6

Posterior Distribution Formulas: (2)

Unfortunately, these are hard to compute directly – *especially when* $\mathbf{x}$ *is high-dimensional*!

Some well-known approximations:

- The Extended Kalman Filter
- Gaussian Sum Filters
- General/Specific Numerical Integration Methods
- Sampling-based methods – Particle Filters.

Monte-Carlo Sampling:

Crude Monte Carlo

The basic idea:

- Estimate the expectation integral

$$I(g) = E_{x \sim p(\mathbf{x})}(g(\mathbf{x})) = \int g(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$$

by the following mean:

$$I_N(g) = \frac{1}{N} \sum_{i=1}^{N} g(\mathbf{x}^{(i)})$$

where $\mathbf{x}^{(i)} \sim p(\mathbf{x})$. That is, $\left\{\mathbf{x}^{(i)}\right\}_{i=1}^{N}$ are independent samples from $p(\mathbf{x})$.

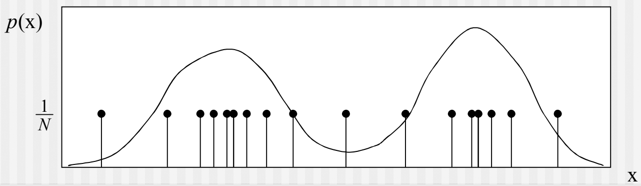
Crude Monte-Carlo – Interpretation

- We can interpret Monte-Carlo sampling as trying to approximate $p(\mathbf{x})$ by the discrete distribution

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

  It is easily seen that $I_N(g) = \int g(\mathbf{x}) \hat{p}_N(\mathbf{x}) d\mathbf{x}.$

- The samples $\mathbf{x}^{(i)}$ are sometimes called "particles", with weights $w^{(i)} = N^{-1}.$



9

Crude Monte Carlo: Properties

The following basic properties follow from standard results for i.i.d. sequences.

1. Bias: $I_N(g)$ is an unbiased estimate of $I(g)$, namely
   $$E(I_N(g)) = I(g).$$
2. SLLN:
   $$I_N(g) \xrightarrow[N\to\infty]{\text{a.s.}} I(g).$$
3. CLT:
   $$\sqrt{N}\,[I_N(g) - I(g)] \xrightarrow[N\to\infty]{\text{in distr.}} \mathcal{N}(0, \sigma_g^2)$$
   whenever
   $$\sigma_g^2 = \operatorname{cov}(g(\mathbf{x})) < \infty.$$

Advantages (over numerical integration):

- Straightforward scheme.
- Focuses on "important" areas of $p(\mathbf{x})$.
- Rate of convergence does not depend (directly) on $\dim(\mathbf{x})$.

10

The Monte-Carlo Method: Origins

- Conceiver: Stanislav Ulam (1946).
- Early developments (1940's-50's):
  - Ulam & Von-Neumann: Importance Sampling, Rejection Sampling.
  - Metropolis & Von-Neumann: Markov-Chain Monte-Carlo (MCMC).

A surprisingly large array of applications, including
- global optimization
- statistical and quantum physics
- computational biology
- signal and image processing
- rare-event simulation
- state estimation …

11

Importance Sampling (IS)

Idea:

- Sample $\mathbf{x}$ from *another* distribution, $q(\mathbf{x})$.
- Use weights to obtain the correct distribution.

Advantages:

- May be (much) easier to sample from q than from p.
- The covariance σ_g^2 may be (much) smaller

  $\Rightarrow$ Faster convergence / better accuracy.

12

Importance Sampling (2)

- Let $q(\mathbf{x})$ be a selected distribution over $\mathbf{x}$ – the *proposal distribution*. Assume that $p(\mathbf{x}) > 0 \Rightarrow q(\mathbf{x}) > 0$.
- Observe that:

$$I(g) = \int g(\mathbf{x})\,p(\mathbf{x})\,d\mathbf{x} = \int g(\mathbf{x})\frac{p(\mathbf{x})}{q(\mathbf{x})}q(\mathbf{x})\,d\mathbf{x}$$
$$= \int g(\mathbf{x})\,w(\mathbf{x})\,q(\mathbf{x})\,d\mathbf{x}$$

where

$$w(\mathbf{x}) = \frac{p(\mathbf{x})}{q(\mathbf{x})}$$

is the likelihood ratio, or simply the *weight-function*.

Importance Sampling (3)

- This immediately leads to the IS algorithm:
  (1) Sample $\mathbf{x}^{(i)} \sim q(\mathbf{x}), \quad i = 1,\ldots,N.$
  (2) Compute the *weights* $w^{(i)} = \frac{p(\mathbf{x}^{(i)})}{q(\mathbf{x}^{(i)})}$, and

$$\hat{I}_N(g) = \frac{1}{N}\sum_{i=1}^{N} w^{(i)} g(\mathbf{x}^{(i)}).$$

Basic Properties:

- The above-mentioned convergence properties are maintained.
- The CLT covariance is now $\hat{\sigma}_g^2 = \mathrm{cov}_{\mathbf{x}\sim q(\mathbf{x})}(g(\mathbf{x})\,w(\mathbf{x}))$.
- This covariance is minimized by $q(\mathbf{x}) = p(\mathbf{x})\,g(\mathbf{x})$. Unfortunately this choice is often too complicated, and we settle for approximations.

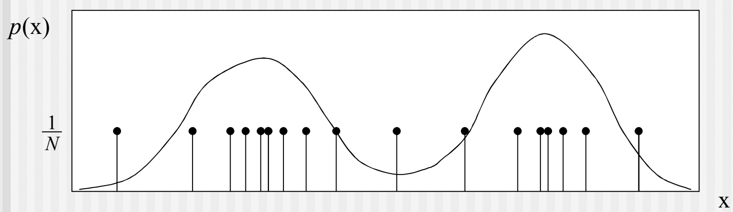
Importance Sampling (4)

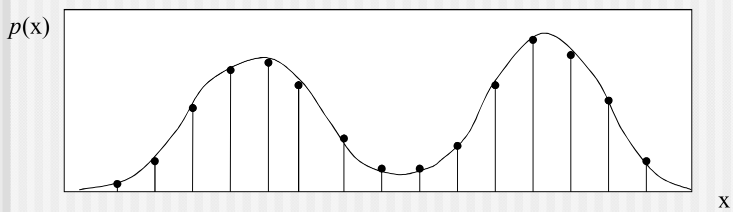
Additional Comments:

- It is often convenient to use the *normalized* weights $\tilde{w}^{(i)} = \frac{1}{N} w^{(i)}$, so that $\hat{I}(g) = \sum_{i=1}^{N} \tilde{w}^{(i)} g(\mathbf{x}^{(i)})$.
- The weights $w^{(i)}$ are sometimes not properly normalized for example, when $p(\mathbf{x})$ is known up to a (multiplicative) constant.
  We can still use the last formula, with $\tilde{w}^{(i)} = \frac{w^{(i)}}{\sum_{j=1}^{N} w^{(i)}}$.
- We may interpret $(\mathbf{x}^{(i)}, \tilde{w}^{(i)})$ as a "weighted-particles" approximation to $p(\mathbf{x})$:

$$p(\mathbf{x}) \approx \hat{p}_N(\mathbf{x}) = \sum_{i=n}^{N} \tilde{w}^{(i)} \delta(\mathbf{x} - \mathbf{x}^{(i)})$$

Two Discrete Approximations





The Particle Filter Algorithm

We can now apply these ideas to our filtering problem. The basic algorithm is as follows:

0. Initialization: sample from $p_0(\mathbf{x})$ to obtain $\{\mathbf{x}_0^{(i)}, w_0^{(i)}\}$, with $w_0^{(i)} = N^{-1}$. For $t=1,2,\ldots,$ given $\{\mathbf{x}_{t-1}^{(i)}, w_{t-1}^{(i)}\}$:
1. Time update: For $i=1,\ldots,N$ sample $\mathbf{x}_t^{(i)} \sim p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(i)})$. Set $\hat{w}_t^{(i)} = w_{t-1}^{(i)}$.
2. Measurement update: For $i=1,\ldots,N$, evaluate the (unnormalized) importance weights: $\overline{w}_t^{(i)} = \hat{w}_t^{(i)} \times p(\mathbf{y}_t \mid \mathbf{x}_t^{(i)})$.
3. Resampling: Sample N points $\{\mathbf{x}_t^{(i)}\}_{i=1}^N$ from the discrete distribution corresponding to $\{\hat{\mathbf{x}}_t^{(i)}, \overline{w}_t^{(i)}\}$. Continue with $\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}$, $w_t^{(i)} = N^{-1}$.

17

The Particle Filter Algorithm (2)

- Output: Based on $\{x_t^{(i)}, w_t^{(i)}\}$, we can estimate

$$p(\mathbf{x}_t \mid \mathbf{y}_{1:t}) \approx \sum_{i=1}^{N} w_t^{(i)} \delta(\mathbf{x} - \mathbf{x}_t^{(i)})$$

$$E(g(\mathbf{x}_t) \mid \mathbf{y}_{1:t}) \approx \sum_{i=1}^{N} w_t^{(i)} g(\mathbf{x}_t^{(i)})$$

We may also use $\{\hat{x}_t^{(i)}, \overline{w}_t^{(i)}\}$ for that purpose.

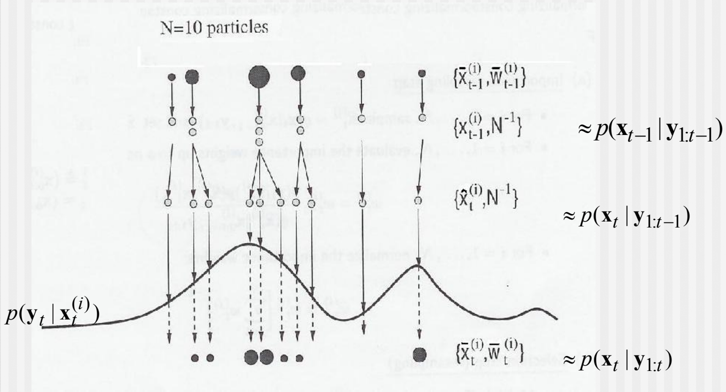
This algorithm is the original "bootstrap filter" introduced in Gordon et al. (1993).

CONDENSATION is a similar algorithm developed independently in the computer vision context (Isard and Blake, 1998).

Many improvements and variations now exist.

18

Particle Flow



(from Doucet et. al., 2001)

The Particle Filter: Time Update

- Recall that $\left\{\mathbf{x}_{t-1}^{(i)}, w_{t-1}^{(i)}\right\}$ approximates $p(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:t-1})$, and $p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ is given.
- An approximation $\left\{\hat{\mathbf{x}}_t^{(i)}, \hat{w}_t^{(i)}\right\}$ is therefore obtained by sampling $\mathbf{x}_t^{(i)} \sim p(\mathbf{x}_t \mid \mathbf{x}_{t-1}^{(i)})$, and setting $\hat{w}_t^{(i)} = w_{t-1}^{(i)}$.
- For example, for the familiar state model $\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{x}_{t-1}),\ v_{t-1} \sim \mathcal{N}(0, Q),$ we get

$$\hat{\mathbf{x}}_t^{(i)} = f(\mathbf{x}_{t-1}^{(i)}, v_{t-1}^{(i)}), \quad v_{t-1}^{(i)} \sim \mathcal{N}(0, Q).$$

The Particle Filter: Time Update (2)

- It is possible (and often useful) to sample $\hat{\mathbf{x}}_t^{(i)}$ from *another* (proposal) distribution $q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$.

  In that case we need to modify the weights:

$$\hat{w}_t^{(i)} = w_{t-1}^{(i)} \frac{p(\mathbf{x}_t^{(i)} \mid \mathbf{x}_{t-1}^{(i)})}{q(\mathbf{x}_t^{(i)} \mid \mathbf{x}_{t-1}^{(i)})}.$$

- The ideal choice for $q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ would by

$$q^*(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \frac{p(\mathbf{x}_t \mid \mathbf{y}_{1:t})}{p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})}.$$

- Some schemes exist which try to approximate sampling with q^* - e.g., using MCMC.

21

The Particle Filter: Measurement Update

- Recall that $p(\mathbf{x}_t \mid \mathbf{y}_{1:t}) = \frac{1}{c} p(\mathbf{y}_t \mid \mathbf{x}_t) p(\mathbf{x}_t \mid \mathbf{y}_{1:t-1})$,

  and $\{\hat{\mathbf{x}}_t^{(i)}, \hat{w}_t^{(i)}\}$ approximate $p(x_t \mid y_{1:t-1})$.

- To obtain an approximation $\{\overline{\mathbf{x}}_t^{(i)}, \overline{w}_t^{(i)}\}_{i=1}^{\mathcal{N}}$ to $p(\mathbf{x}_t \mid \mathbf{y}_{1:t})$, we only need to modify the weights accordingly:

$$\overline{w}_t^{(i)} = \hat{w}_t^{(i)} p(\mathbf{y}_t \mid \hat{\mathbf{x}}_t^{(i)}),$$

  with $\overline{\mathbf{x}}_t^{(i)} = \hat{\mathbf{x}}_t^{(i)}$.

22

The Particle Filter: Resampling

- Resampling is not strictly required for asymptotic convergence (as $N \to \infty$).
- However, without it, the distributions $\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}$ tend to degenerate, with few "large" particles, and many small ones.
- The idea is therefore to split large particles into several smaller ones, and discard very small particles.
- Random sampling from $\{\bar{\mathbf{x}}_t^{(i)}, \bar{w}_t^{(i)}\}$ is one option. Another is a deterministic splitting of particles according to $w_t^{(i)}$, namely: $(\bar{\mathbf{x}}_t^{(i)}, \bar{w}_t^{(i)}) \to N_i \approx \lfloor N\, w_t^{(i)} \rfloor$ particles at $\mathbf{x}_t^{(i)} = \bar{\mathbf{x}}_t^{(i)}$, with $w_t^{(i)} = N_i^{-1} \bar{w}_t^{(i)}$.

  In fact, the last option tends to reduce the variance.

Additional Issues and Comments

Data Association:

- This is an important problem for applications such as multiple target tracking, and visual tracking.
- Within the Kalman Filter framework, such problems are often treated by creating multiple explicit hypothesis regarding the data origin which may lead to combinatorial explosion.
- Isard and Blake (1998) observed that particle filters open new possibilities for data association for tracking. Specifically, different particles can be associated to different measurements, based on some "closeness" (or likelihood) measure.
- Encouraging results have been reported.

Additional Issues and Comments (2)

Joint Parameter and State Estimation:

- As is well known, this "adaptive estimation" problem can be reduced to a standard estimation problem by embedding the parameters in the state vector.
- This however leads to a bi-linear state equation, for which standard KF algorithms did not prove successful. The problem is currently treated using non-sequential algorithms (e.g. EM + smoothing).
- Particle filters have been applied to this problem with encouraging results.

25

Additional Issues and Comments (3)

Low-noise State Components:

- State components which are not excited by external noise (such as fixed parameters) pose a problem since the respective particle components may be fixed in their place, and cannot correct initially bad placements.
- The simplest solution is to add a little noise. Additional suggestions involve intermediate MCMC steps, and related ideas.

26

Conclusion:

- Particle Filters and Sequential Monte Carlo Methods are an evolving and active topic, with good potential to handle "hard" estimation problems, involving non-linearity and multi-modal distributions.
- In general, these schemes are computationally expensive as the number of "particles" N needs to be large for precise results.
- Additional work is required on the computational issue – in particular, optimizing the choice of N, and related error bounds.
- Another promising direction is the merging of sampling methods with more disciplined approaches (such as Gaussian filters, and the Rao-Blackwellization scheme …).

To Probe Further:

Monte-Carlo Methods (at large):

- R. Rubinstein, *Simulation and the Monte-Carlo Method*, Wiley, 1981 (and 1996).
- J.S. Liu, *Monte-Carlo Strategies in Scientific Computing*, Springer, 2001.
- M. Evans and T. Swartz, "Methods for approximating integrals in statistics with special emphasis on Bayesian integration problems", *Statistical Science* 10(3), 1995, pp. 254-272.

Particle Filters:

- N.J. Gordon, D. J. Salmond and A. Smith, "Novel approach to non-linear / non-Gaussian Bayesian state estimation", *IEE Proceedings-F* 140(2), pp. 107-113.
- M. Isard and A. Blake, "CONDENSATION – conditional density propagation for visual tracking", *International Journal of Computer Vision* 28(1), pp. 5-28.
- A Doucet, N. de Freitas and N. Gordon (eds.) *Sequential Monte Carlo Method in Practice*, Springer, 2001.
- S. Arulampalam et al., "A tutorial on particle filters, for on-line non-linear/non-Gaussian Bayesian tracking*", IEEE Trans. on Signal Processing* 50(2), 2002.