Importance Sampling

Elaine Spiller

Marquette University

ICTS Workshop on Nonlinear Filtering & Data Assimilation

January 9, 2014

This talk

- Basic overview of MC simulations & importance sampling
- From simulation/implementation/algorithmic point of view
- References:
  - *Simulations*, Ross
  - *Monte Carlo Statistical Methods*, Robert and Casella
  - *Introduction to Rare Event Simulation*, Bucklew
- Basic overview of particle filtering algorithm and use of importance sampling
- Apologies for abuses of probability notation (notation from engineering/stat methods lit)
- References:
  - Lui and Chen, *JASA* 1998 (and references there in)
  - A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice* 2001
- Application(s) from Lagrangian Data Assimilation

Monte Carlo Methods

Suppose $X_1, X_2, \ldots$ are iid random variables taken from a distribution $f(x)$. Given a function $g(x)$, define

$$G = \frac{1}{N} \sum_{n=1}^{N} g(X_n)$$

$$E[G] = E[\frac{1}{N} \sum_{n=1}^{N} g(X_n)] = \frac{1}{N} \sum_{n=1}^{N} E[g(X)] = E[g(X)]$$

And

$$var\{G\} = \sum_{n=1}^{N} \frac{1}{N^2} var\{g(X)\} = \frac{1}{N} var\{g(X)\}$$

So, as $N \to \infty$, $var\{G\} \to 0$.

Monte Carlo Integration

Recall, $$E_f[g(x)] = \int_{-\infty}^{\infty} g(x)f(x)dx$$

And, we just saw $E[G] = E[g]$, so we have the basis for Monte Carlo integration

$$\int_{-\infty}^{\infty} g(x)f(x)dx \approx \frac{1}{N}\sum_{n=1}^{N} g(X_n) \qquad \text{where } X_n \sim f.$$

Another common case: $g(x) = 1$

$$P[X > a] = \int_{a}^{\infty} f(x)dx \approx \frac{1}{N}\sum_{n=1}^{N} I(X_n > a)$$

How to justify this

Law of Large Numbers:

$$\text{If} \quad \bar{X}_N = \frac{1}{N}\sum_{n=1}^{N} X_n, \quad \text{then} \quad P\{\lim_{N\to\infty} \bar{X}_N = E[X]\} = 1$$

$$\text{So,} \quad G \xrightarrow[n\to\infty]{} \int_{-\infty}^{\infty} g(x)f(x)dx$$

If we take enough samples, Monte Carlo (MC) integration will **always** converge.

Question: How much is *enough*?

$$P\Big\{|G - E[G]| \geq \Big[\frac{var\{G\}}{\delta}\Big]^{1/2}\Big\} \leq \delta$$

This could be called the Fundamental Theorem of Monte Carlo methods because it estimates the probably of a large deviation of an MC calculation.

Rewriting, we have

$$P\Big\{\Big(\frac{1}{N}\sum_{n=1}^{N} g(X_n) - \int_{-\infty}^{\infty} g(x)f(x)dx\Big)^2 \geq \frac{var\{g(X)\}}{\delta N}\Big\} \leq \delta.$$

In words, this says that the probability that a sample calculation & the exact solution differ by $\sqrt{\frac{1}{\delta N} var\{g(x)\}}$ is no more than δ.

For a specified "certainty", δ, there are only two ways to manipulate the magnitude of the error

$$\text{error}^2 = \frac{var\{g(X)\}}{\delta N}$$

- increase the number of MC samples, N
- decrease $var\{g(X)\}$

If you think about MC as a method to calculate an integral

$$\int_a^b h(x)dx,$$

you get to choose how to "break up" h into $h(x) = g(x)f(x)$.

Monte Carlo example

How many samples, N, must we take to have a 0.99 probability of calculating the integral

$$\int_0^2 x^3 dx$$

withing an error of 0.1?

Note, probability (certainty) of 0.99 means $\delta = 0.01$.

To do this, we need to write $x^3 = g(x)f(x)$ where $\int_0^2 f(x)dx = 1$.

Let's do two cases

1. $f(x)$ Uniform, so $f(x) = \frac{1}{2} \quad 0 < x < 2$ and $g(x) = 2x^3$
2. $f(x) = \frac{3}{8}x^2$ and $g(x) = \frac{8}{3}x$

MC example continued

Really what we are asking for is

$$\frac{1}{\delta N} var\{g(x)\} < \text{error}^2 < \left(\frac{1}{10}\right)^2.$$

So, we need to calculate $var\{g(x)\}$ and solve for N.

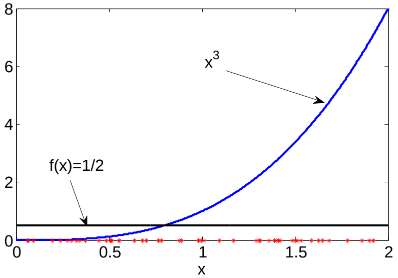
For our two cases, we'll see

1. $N \approx 20x10^4$
2. $N \approx 1x10^4$

Question:

Why is this happening?

What's going on in MC example



Moral of the story:

We want to sample the domain where the integrand is big.

The magnitude of the variance of $g(X)$ in some sense tells us "how well" we're accomplishing that goal.

Pros and cons of MC

$$\text{error}^2 = \frac{var\{g(X)\}}{\delta N}$$

Cons:

- quite inefficient compared to grid-based methods, $O(1/\sqrt{N}) = O(\sqrt{\Delta x})$
- $f(x)$ can be hard to sample

Pros:

- error is *independent* of dimension
- we have some freedom to reduce $var\{g(X)\}$
- natural framework for inherently stochastic problems

Importance sampling: why?

Recall, the key to efficient Monte Carlo algorithms is a reduction in the variance of the estimator,

$$G = \frac{1}{N}\sum_{n=1}^{N} g(X_n)$$

The greater the variance, the more sample points, N, needed to (accurately) estimate the quantity of interest.

Suppose we wish to integrate

$$E_f[g] = \int_{-\infty}^{\infty} g(x) f(x) dx,$$

we could estimate this with samples from *any* probability distribution we like.

Importance sampling: idea

Let's consider $\tilde{f}(x) > 0$ and rewrite our integral as

$$E_f[g] = \int_{-\infty}^{\infty} \frac{g(x)f(x)}{\tilde{f}(x)} \tilde{f}(x) dx.$$

At this point, we could sample from $f(x)$ or $\tilde{f}(x)$, but if we sample $\tilde{f}(x)$, then

$$\tilde{g}(x) = \frac{g(x)f(x)}{\tilde{f}(x)}$$

and the MC error is determined by

$$var\{\tilde{g}\} = \int_{-\infty}^{\infty} \left[\frac{g^2(x)f^2(x)}{\tilde{f}^2(x)}\right] \tilde{f}(x) dx - E_{\tilde{f}}^2[\tilde{g}].$$

Note, $E_{\tilde{f}}[\tilde{g}] = E_f[g] =$ some (unknown) constant. And recall, we'd like to minimize $var\{\tilde{g}\}$.

Importance sampling: how to choose $\tilde{f}$?

To minimize $var\{\tilde{g}\}$, we want an $\tilde{f}$ such that

$$\int_{-\infty}^{\infty} \left[\frac{g^2(x)f^2(x)}{\tilde{f}^2(x)}\right]\tilde{f}(x)dx \quad \text{is minimized, subject to} \quad \int_{-\infty}^{\infty} \tilde{f}(x)dx = 1.$$

This can be solved via Lagrange multipliers, resulting in an optimal $\tilde{f}$ of

$$\tilde{f}(x) \propto g(x)f(x)$$

- Does this result seem reasonable?
- Is it useful???

Optimal biasing distribution, $\tilde{f}$

Is $\tilde{f} \propto g(x)f(x)$ reasonable ?

- Yes, choosing this $\tilde{f}$, we get $var\{\tilde{g}\} = 0$. This is the best we can do!
- No, to make $\tilde{f}$ a distribution, we need to normalize it by

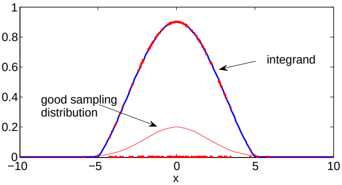
$$\tilde{f}(x) = \frac{f(x)g(x)}{\int_{-\infty}^{\infty} f(x)g(x)dx}.$$

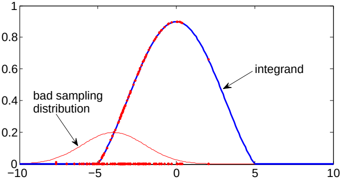
  But that normalization factor is exactly the integral we'd like to approximate!

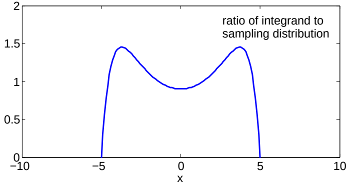
Is $\tilde{f} \propto g(x)f(x)$ useful?

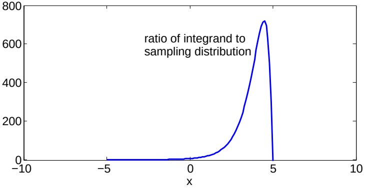
- Yes, the general goal is to make the importance distribution $\tilde{f}$ "look like" the integrand, $f(x)g(x)$

Good vs.bad importance distribution









Importance sampling example

Consider the integral

$$E[g(x)] = \int_0^1 \cos(\frac{\pi}{2}x)dx$$

If we pick $\tilde{f}$ to be $U(0,1)$, the we get a variance of

$$\int_0^1 \cos^2(\frac{\pi}{2}x)dx - E^2[g] \approx 0.09472$$

Recall, we want $\tilde{f}(x) \approx \cos(\frac{\pi}{2}x)$, we could choose a $\tilde{f}$ by expanding

$$\cos(\frac{\pi}{2}x) = 1 - \frac{\pi^2}{8}x^2 + \frac{\pi^4}{2^4 4!}x^4 + \dots$$

This suggests
$\tilde{f}(x) = \alpha(1 - \frac{\pi}{8}x^2)$ with α chosen so $\int_0^1 (1 - \frac{\pi}{8}x^2)dx = 1/\alpha$

IS example continued

Problem, $\tilde{f}(x) = \alpha(1 - \frac{\pi}{8}x^2) < 0$ on $0 < x < 1$.

So let's try

$$\tilde{f}(x) = \alpha(1 - x^2) \qquad \text{with } \alpha = 3/2.$$

Computing the variance for the resulting $\tilde{g}$, we get

$$\int_0^1 \frac{\cos^2(\frac{\pi}{2}x)}{\frac{3}{2}(1 - x^2)} dx - \frac{4}{\pi^2} \approx 0.000990$$

So by choosing a better $\tilde{f}$, we have succeeded in reducing that variance of our estimator by a factor of 100.

How to implement IS in practice

Recall, we are looking to estimate

$$E_f[g] = \int_{-\infty}^{\infty} g(x)f(x)dx$$

In practice, we sample X_n from $\tilde{f}$ and we have an estimate to the mean of g,

$$E_f[g] = E_{\tilde{f}}[\tilde{g}] \approx \frac{1}{N}\sum_{n=1}^{N} g(X_n)\frac{f(X_n)}{\tilde{f}(X_n)}.$$

$$\frac{f(X_n)}{\tilde{f}(X_n)} \quad \text{is called the likelihood ratio.}$$

In some sense it tells us how likely each realization of X_n would have been if it had come from f instead of $\tilde{f}$.

Note on IS in multiple dimensions

Suppose g is a function of many random variables, say $\mathbf{Y} = (X_1, \ldots, X_k)$ and the $X_k's$ are independent. In this case

$$f(\mathbf{Y}) = \prod_{i=1}^{k} f_i(X_i) \quad \text{and similarly for } \tilde{f}(\mathbf{Y}), \;\; \text{so}$$

$$\frac{f(\mathbf{Y})}{\tilde{f}(\mathbf{Y})} = \prod_{i=1}^{k} \frac{f_i(X_i)}{\tilde{f}_i(X_i)}$$

So in words, the likelihood ratio is the product of individual likelihood ratios.

(Note, in practice products of “small things” are often numerically unstable.)

Another motivation for importance sampling: Rare Events

Recall MC for our second "type" of problem

$$P(X > a) = \int_a^\infty f(x)dx \approx \frac{1}{N}\sum_{n=1}^{N} I(X_n > a) \quad X_n \sim f$$

- Guaranteed to converge by the Law of Large Numbers.
- In practice if $a >> \mu$, it will **not** converge.
- A *rare event* is lousy defined to have $P \leq 10^{-6}$

Using importance sampling, we have

$$P(X > a) = \int_a^\infty f(x)dx \approx \frac{1}{N}\sum_{n=1}^{N} I(X_n > a)\frac{f(X_n)}{\tilde{f}(X_n)}$$

with $X_n \sim \tilde{f}$.

Simplest example: 100 coin flips

question: What is P(70 or more heads) ?

answer: $2.4x10^{-13}$

importance sampling: Use weighted coin

$$\begin{aligned} \tilde{p} &= 0.7 \qquad \text{for heads} \\ \tilde{p} &= 0.3 \qquad \text{for tails} \end{aligned}$$

likelihood ratios for flipping weighted coin:

$$\frac{p}{\tilde{p}} = \begin{cases} 0.5/0.7 & \text{for heads} \\ 0.5/0.3 & \text{for tails} \end{cases}$$

key: correcting with likelihood ratio gives statistics for fair coin

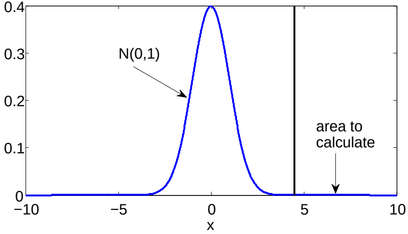
efficiency: over 10 orders of magnitude speed-up

Another rare event example

Say $Z \sim N(0,1)$ and we are interested in $P(Z > 4.5)$. Approximating this with MC gives us

$$P(Z > 4.5) \approx \frac{1}{N}\sum_{n=1}^{N} I(Z_n > 4.5). \qquad Z_n \sim \tilde{f} = N(0,1)$$

Typically $N = 10,000$ samples produces *all zeros* of the indicator function.

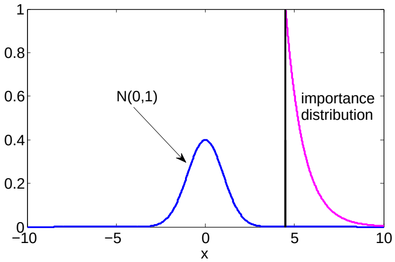


Instead use a shifted Exponential distribution.

Example continued

A good shifted exponential would be

$$\tilde{f}(x) = \frac{e^{-(x-4.5)}}{\int_{4.5}^{\infty} e^{-(x-4.5)}dx} \qquad \text{for } x > 4.5$$



Now if $X \sim \tilde{f}$

$$P(Z > 4.5) \approx \frac{1}{N}\sum_{n=1}^{N}\frac{f(X_n)}{\tilde{f}(X_n)} = 0.000003377$$

Particle filters – pieces

We're given:

background distribution of initial conditions: $p^f(x_0)$

dynamic model: $x_j = M(x_{j-1})$

associated transition density: $m(x_j|x_{j-1})$

an observation operator: $y = h(x)$

observations: $y_{1:n}$

observation noise model: g

Note, $$p^f(x_1) = \int m(x_1|x_0)p^f(x_0)dx_0$$

Particle filter – Bayes Theorem

- Prior/forecast

$$p^f(x_{0:n}) = p^f(x_0)\prod_{j=1}^{n} m(x_j|x_{j-1})$$

- Likelihood

$$p(y_{1:n}|x_{1:n}) = \prod_{j=1}^{n} g(y_j = h(x_j)|x_j)$$

- Posterior/analysis, obtained by Bayes' rule

$$p^a(x_{1:n}|y_{1:n}) = \frac{p(y_{1:n}|x_{1:n})p^f(x_{0:n})}{p(y_{1:n})}$$

recall, $p(y_{1:n}) = \int p(y_{1:n}|x_{1:n})p^f(x_{0:n})dx_{1:n}$

Particle filter – One step

- Prior/forecast

$$p^f(x_{0:1}) = p^f(x_0)m(x_1|x_0)$$

$$p^f(x_1) = \int m(x_1|x_0)p^f(x_0)dx_0$$

- Posterior/analysis, obtained by Bayes' rule

$$p^a(x_1|y_1) = \frac{p(y_1|x_1)p^f(x_1)}{\int p(y_1|x_1)p^f(x_1)dx_1}$$

Particle filter – one step algorithm

Think of a "particle" as a state variable w/a weight attached, $\{x^i, w^i\}$ $i = 1, \ldots, N_{part}$ which approximates a pdf

- sample $X_0^i \sim p^f(x_0)$, $i = 1, ..., N_{part}$
- set $x_0^i = X_0^i$
- push forward $x_1^i = M(x_0^i)$

Now $\{x_1^i, 1/N_{part}\} \approx p^f(x_1)$

$$p^a(x_1|y_1) = \frac{p(y_1|x_1)p^f(x_1)}{\int p(y_1|x_1)p^f(x_1)dx_1}$$

- find $w_1^i = p(y_1|x_1^i)/\sum_i^{N_{part}} p(y_1|x_1^i)$

So, $\{x_1^i, w_1^i\} \approx p^a(x_1|y_1)$

Particle filter – Sequential Monte Carlo

A Monte Carlo simulation or really sampling $p^a(x_{1:n}|y_{1:n})$ or $p^a(x_n|y_{1:n})$

- takes a discrete set of samples from $X_0^i \sim p(x_0)$, let $x_0^i = X_0^i$
- moves them forward accord to the model, e.g. samples $x_1^i = M(x_0^i)$
- evaluates likelihood between samples and observations

Note, after a few (say $k = 2$ or 3 observations) you will have samples from $X_{0:k} \sim p^a(x_{0:k}|y_{1:k})$. Or, marginalized, $X_k \sim p^a(x_k|y_{1:k})$ namely $\{x_3^i, w_3^i\}$ but they will not be useful.

Sequential Monte Carlo with Importance Sampling (SIS)

Idea — normalize at every step, treat that posterior distribution as an *importance* prior distribution for the next step. That is,

1. Start with $X_0 \sim p(x_0)$, each particle $x_0^i = X_0^i$ has weight $w_0^i = 1/N_{part}$
2. Transition each x_o^i forward, this gives $x_1^i = M(x_0^i)$ & $\{x_1^i, w_0^i\}$
3. Evaluate the likelihood function of each sample ("particle") x_1^i against y_1, $p(y_1|x_1^i)$
4. *Weight* each particle by

$$w_1^i = \frac{p(y_1|x_1^i)w_0^i}{\sum_{i=1}^{N_{part}} p(y_1|x_1^i)w_0^i}$$

5. Repeat, $x_2^i = M(x_1^i)$, $\{x_2^i, w_1^i\} \approx p^f(x_2^i|y_1)$ and

$$w_2^i = \frac{p(y_2|x_2^i)w_1^i}{\sum_{i=1}^{N_{part}} p(y_2|x_2^i)w_1^i}, \quad \text{and} \quad \{x_2^i, w_2^i\} \approx p^a(x_2|y_{1:2})$$

6. Repeat,...

Problem with SIS and Solution: Resampling

With a large number of samples, SIS works pretty well on moderate (small) dimensional deterministic (perfect model) problems.

Problem:

- A significant problem, though, is that most (or all) of the weight can be taken over by *one particle*

Solution:

- Resampling, e.g., bootstrapping

Sequential Importance Recampling

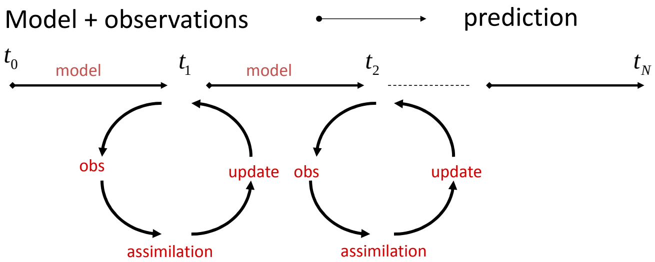
Strategy:

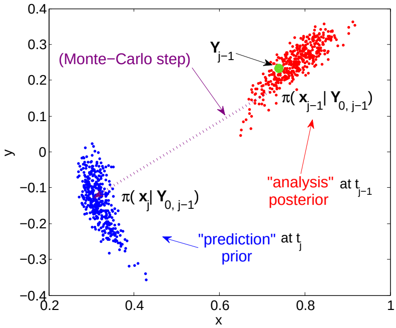
- Monitor weights, if problematic
- *Resample* or "bootstrap" by treating $\{x_k^i, w_k^i\} \approx p^a(x_k|y_{1:k})$ as an *importance* empirical distribution
- Set all weights to $w_k^i = 1/N_{part}$
- Transition $k+1$ step, repeating resampling as necessary

The strategy is referred to an *SIR (sequential importance resampling) filter* and also goes by the names *particle filter*, *bootstrap filter*, and *sequential Monte Carlo*.

(Note, if model is deterministic, need some strategy to sample "around" each x_k^i)

Particle filters: from t_{j-1} to t_j





<u>discrete approx:</u>

Particles are the support of the discrete approximations to these distributions

Each particle is associated with a weight, $w_j(x_j)$

Particle filters: update/analysis at $t = t_j$

Know (discrete approximation):

$$\pi(x_j | Y_{0,j-1}) \text{ (from last page)}$$

Bayes:

$$\pi(x_j | Y_{0,j}) \propto g(Y_j | x_j)\pi(x_j | Y_{0,j-1})$$

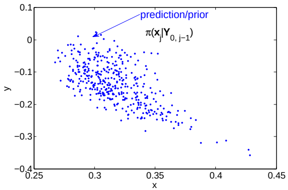
Likelihood:

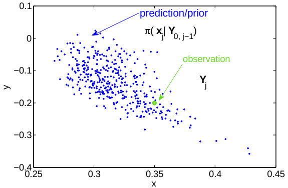
$$g(Y|x) = \exp\left[\frac{H(x) \cdot Y}{\theta^2} - \frac{|H(x)|^2}{2\theta^2}\right]$$

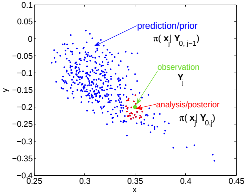
Update (discrete Bayes):

$$w_j(x_j) \propto g(Y_j | x_j) w_j^p(x_j)$$

$$\pi(x_j | Y_{0,j}) = \{x_j, w_j(x_j)\}$$







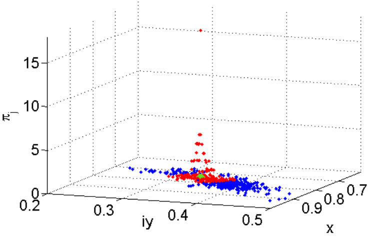
Resampling

problem: degeneracy

- all the weight gets centered on a few particles
- well known and studied

idea:

- pick subset of "best" particles $k = 1, \ldots, M$
- make m_k copies of each particle where $m_k \propto w_j(x_j^{(k)})$ where $\sum m_k = N_{part}$
- prior weight of resampled cloud is $1/N_{part}$



reasonable:

- doesn't add sampling error
- stochastic evolution to t_{j+1} "spreads out" cloud

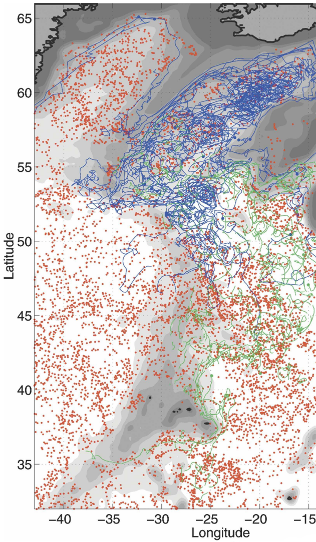
Lagrangian data assimilation (LaDA)

We have available with us the following

- Observations from floats or drifters
  - positions of the drifters
  - prognostic variables such as temperature and salinity of ocean at the location of the drifter
- A model for the velocity field and the other coupled dynamical variables (temperature, salinity, etc.)
- A model for the drifters (typically Lagrangian)

We are mainly interested in

- estimate of the prognostic variables
- positions of the drifters



from Lankhorst, Zenk (2006) JPO,**36**

Skew-product structure of the LaDA problem

An approach to assimilating the information about the observations of positions of drifters is to combine

- the prognostic variables (collectively denoted by x^F) and
- the positions of the drifters (denoted by x^D)

into the state vector: $x = (x^F, x^D)^T$

The model has a skew-product structure,

$\dot{x} = M(x) = (M_v(x), M_d(x))^T$, in the case of passive drifters (which is what we will assume):

$$\frac{dx^F}{dt} = M_v(x^F)\,, \qquad \frac{dx_d}{dt} = M_d(x^F, x^D) = V(x^F, x^D)\,,$$

where V is the velocity of the fluid flow at the point x^D.

Originated and studied extensively in the work of Ide, Jones, Kuznetsov, Salman, ...

Observation operator is the projection on x^D variables

- If only the drifter locations are observed with noise, then the observations at time t can be written as

  $$y(t) = Hx(t) + \eta$$

  where $x = (x^F, x^D)$, and thus $H = [0 \;\; I]$ is just a projection.

- Observation operator is linear in this case

- These observations contain information about the velocity flow as well.

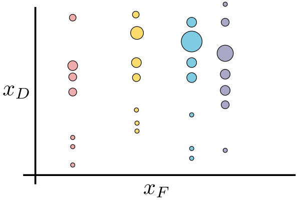
Note: EnKF performance on LaDA degrades as observation period grows

Hybrid PF-EnKF

- EnKF on high-dimensional flow state x^F
- PF on low-dimensional, highly nonlinear Lagrangian coordinates x^D

Ensemble:

$$\{x_i^F, x_{i,j}^D, w_{i,j}\}_{i=1\ldots N_e,\, j=1\ldots M}$$



Update weights via standard particle filter update, and at resampling times, update x^F according to EnKF analysis.

- Update step (no resampling): Treat as typical particle filter and update weights using observation (no updates on particles or ensemble members):

$$w_{i,j}^{new} \propto w_{i,j}^{old} p(y|x_{i,j}^{D})$$

- Update/resampling step: prior $\{x_i^{F,f}, x^{D,f}i, w_{i,j}^{old}\}$
  - Update weights as above. Resample drifter particles from conditional distribution $p(x^D|x^F, y)$.
  - “Resample” flow members from EnKF distribution: perform EnKF update on flow members, using weighted prior covariances. $x^{F,a}$. So, flow “particles” are given by $\{x^{F,a}, w_i^{old}\} \approx p^a(x^F|y)$. Resample this distribution.

Results: Drifter crossing between cells

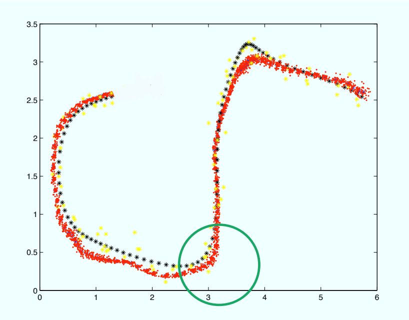


Figure: Ensemble Kalman filter

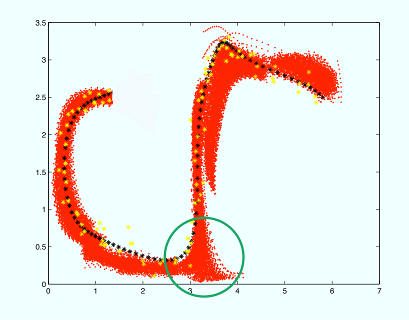
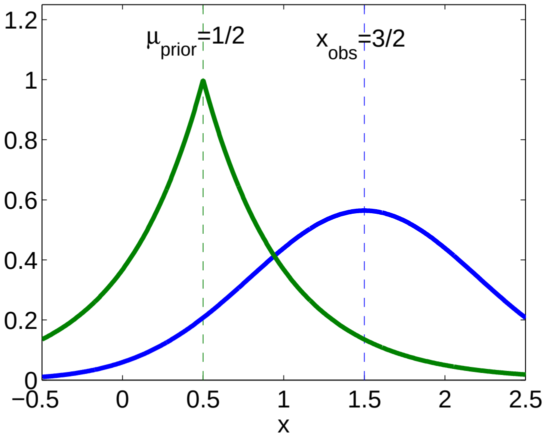


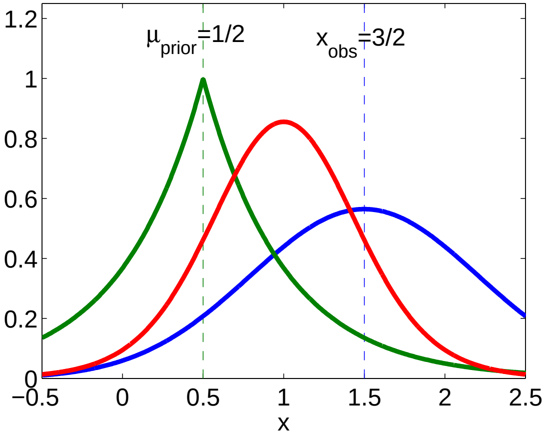
Figure: Hybrid filter

Updating w/nonlinear prior: toy example



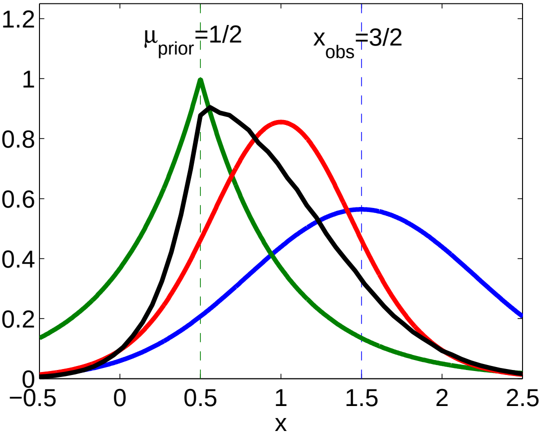
- weighted samples from prior $\{w_i, x_i\}$
- prior and noise model same variance

Updating w/non-Gaussian prior: toy example



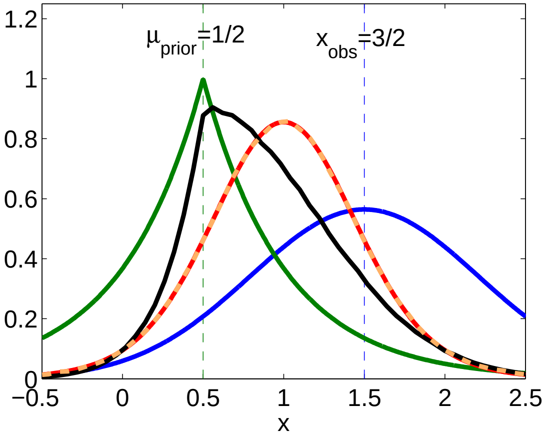
- Kalman filter analysis would give us the convolution of the prior and noise model

Updating w/non-Gaussian prior: toy example



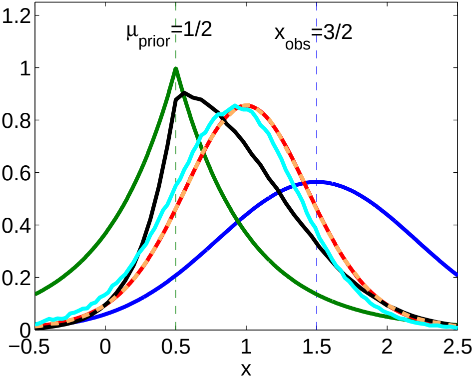
- **posterior** from Bayes Rule

Updating w/non-Gaussian prior: toy example



- translate x to x^a using observation via Kalman Filter
- posterior samples $\{w_i^{old}, x_i^a\}$, mean μ_a and variance σ_a^2

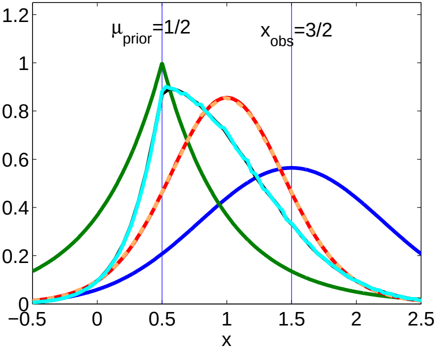
Updating w/non-Gaussian prior: toy example



- reweight samples to get $\{w_i^{new}, x_i^a\}$ that would give you first to moments of non-Gaussian posterior, μ_{pos}, σ^2_{pos}

$$w_i^{new} = w_i^{old} \cdot \frac{\exp\left(-(x_i^a - \mu_{pos})^2/2\sigma^2_{pos}\right)}{\exp\left(-(x_i^a - \mu_a)^2/2\sigma^2_a\right)}$$

Updating w/non-Gaussian prior: toy example



- reweight samples to get $\{w_i^{new}, x_i^f\}$ that would give you first to moments Kalman (Gaussian) posterior, μ_a, σ_a^2

$$w_i^{new} = w_i^{old} \cdot \frac{\exp\left(-(x_i^f - \mu_a)^2/2\sigma_a^2\right)}{\exp\left(-(x_i^f - \mu_f)^2/2\sigma_f^2\right)}$$