INFORMS TutORials in Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Particle Methods for Data-Driven Simulation and Optimization

John R. Birge

To cite this entry: John R. Birge. Particle Methods for Data-Driven Simulation and Optimization. *In* INFORMS TutORials in Operations Research. Published online: 14 Oct 2014; 92-102.
https://doi.org/10.1287/educ.1120.0104

TUTORIALS IN
OPERATIONS RESEARCH
INFORMS 2012

 | ISBN 978-0-9843378-3-5
http://dx.doi.org/10.1287/educ.1120.0104

Particle Methods for Data-Driven Simulation and Optimization

John R. Birge
The University of Chicago Booth School of Business, Chicago, Illinois 60637,
john.birge@ChicagoBooth.edu

Abstract Particle methods provide a robust methodology for estimation and prediction. They can also apply to traditional operations research areas by, for example, providing an effective alternative for constructing Monte Carlo simulations of various systems and for dynamic optimization. This tutorial explains the fundamental ideas behind particle methods and how to use them in typical operations research contexts.

Keywords particle methods; simulation; optimization

1. Introduction

Particle methods have become a common approach to estimate the distribution of an unknown state in a Markov process (i.e., a hidden Markov model) from noisy observations with general nonlinear transitions. Whereas these techniques have considerable value for estimation, they can also inform forward-looking operations research (OR) problems such as simulation and optimization of complex systems. This tutorial provides an elementary introduction to the topic and focuses on applications of simulation and optimization that are common in the OR community.

Many other tutorials explain particle methods in the context of estimation. Examples of these surveys include those by Andrieu and Doucet [3], Arulampalam et al. [4], Cappé et al. [7], and Doucet and Johansen [11]. These surveys discuss various alternative strategies as well as some of the theoretical developments in the methodology. In contrast, this tutorial aims only to introduce the topic generally and to demonstrate how the basic methodology can apply to typical situations in operations research. In particular, this paper aims to motivate particle method and sequential Monte Carlo approaches for exploring rare event probabilities and optimizing systems.

The tutorial begins by introducing particle methods as a basic estimation method in §2. Section 3 then presents an example for estimating inventory position from noisy observations. Section 4 then presents a broader view of particle methods that can be applied for sequential importance sampling (SIS) in dynamic simulation experiments where standard Monte Carlo methods may have difficulties, such as exploring tail behavior. Section 5 then presents an extension for optimization. Section 6 presents conclusions.

2. Basic Method

The origin of particle methods is often given as Gordon et al. [13]. The discussion here essentially follows that development. The objective is to obtain an estimate of the distribution X_t of a Markov process at a sequence of times $t = 1, \ldots, T$ from a set of noisy observations Y_t. The initial state X_1 is assumed to have a known distribution $\mu(x_1)$ for an element x_1, and the distribution of transitions from X_{t-1} to X_t is given by $f(x_t \mid x_{t-1})$, which is also known,

as is $g(y_t \mid x_t)$, the distribution of Y_t given X_t. For this exposition, we assume f and g are stationary, although a general framework allows for time dependence. The fundamental issue is that although the transitions and conditional distributions of the observations are known, the distribution of X_t conditional on observation realizations $y_1, \ldots, y_t$ may not have a simple representation even when μ, f, and g belong to well-known distribution families. *Particle filtering* methods provide a discrete representation of this distribution through the use of a finite set of samples (particles). The related *smoothing* problem (with associated *particle smoothing methods*) refers to finding the distribution of X_t given all process observations $y_1, \ldots, y_T$.

This framework defines a *hidden Markov model* in which the true underlying process is not directly observed and must only be inferred from the observations. The goal in filtering is to find a distribution $p(x_1, \ldots, x_t \mid y_1, \ldots, y_t)$ (or, in more compact notation, $p(x^t \mid y^t)$, where $x^t = (x_1, \ldots, x_t)$ and $y^t = (y_1, \ldots, y_t)$), which can be evaluated in a Bayesian framework starting with the prior μ on x_1 as

$$p(x^t \mid y^t) = p(x^t, y^t) \Big/ \int p(y^t \mid x^t)\, dx^t, \tag{1}$$

which can then be used to find

$$p(x_t \mid y^t) \propto \int g(y_t \mid x_t) \int f(x_t \mid x_{t-1}) p(x^{t-1}, y^{t-1})\, dx^{t-1}, \tag{2}$$

where the proportionality refers to the problematic constant of integration (which, in (2), is $p(y_t \mid y^{t-1})$). The property exploited in this and other sequential Monte Carlo methods is that Markov draws proportional to the mixing density converge to the true distribution without requiring explicit knowledge of it. By repeatedly applying the observation in (2), the fundamental relationship is then that

$$p(x_t \mid y^t) \propto \mu(x_1) g(y_1 \mid x_1) \prod_{s=1}^{t-1} f(x_{s+1} \mid x_s) g(y_{s+1} \mid x_{s+1}). \tag{3}$$

The essence of the algorithm is to apply (3) sequentially to samples of x_{s+1} using previous samples of x_s and then to weight these samples by $g(y_{s+1} \mid x_{s+1})$. The basic algorithm also incorporates a resampling (or *bootstrapping*) step in which the samples are redrawn whenever they become too concentrated (i.e., have excessive weight on a small number of particles). The particle filtering method then proceeds as follows with an independent and identically distributed draw $x_{1,1}, \ldots, x_{1,N}$ with weights $w_{0,j} = 1/N$, $j = 1, \ldots, N$, from using $p(x_1 \mid y_1)$, which is given by

$$p(x_1 \mid y_1) = \mu(x_1) g(y_1 \mid x_1) \Big/ \int \mu(x_1) g(y_1 \mid x_1)\, dx_1, \tag{4}$$

and the following general steps at each $t = 1, \ldots, T-1$.

Particle filtering method: Sequential importance resampling (SIR).

(1) *Propagate.* Draw $x_{t+1,j}$, $j = 1, \ldots, N$ from $f(x_{t+1,j} \mid x_{t,j})$.

(2) *Reweight.* Update weights, $\hat{w}_{t,j}$, $j = 1, \ldots, N$, to $\hat{w}_{t+1,j} = (w_{t,j} g(y_{t+1} \mid x_{t+1})) / (\sum_{j=1}^{N} \hat{w}_{t,j} g(y_{t+1} \mid x_{t+1}))$ for $j = 1, \ldots, N$.

(3) *Resample.* If a resampling condition is satisfied, draw N new samples $x'_{t+1,j}$, $j = 1, \ldots, N$ from $x_{t+1,j}$, $j = 1, \ldots, N$ with probability distribution given by $\hat{w}_{t+1,j}$, $j = 1, \ldots, N$; let $x_{t+1,j} = x'_{t+1,j}$ and $\hat{w}_{t+1,j} = 1/N$ for $j = 1, \ldots, N$.

This basic algorithm can be generalized in a number of ways. First, the algorithm above uses $f(x_{t+1} \mid x_t)$ for propagation and reweighting, although generally this could include dependence on y_{t+1} by, for example, explicitly drawing from $g(y_{t+1} \mid x_{t+1}) f(x_{t+1} \mid x_t) / p(y_{t+1} \mid x^{t+1})$, which would generally be too difficult to find. Instead, the algorithm can

use some approximation $q(x_{t+1} \mid y_{t+1}, x^{t+1})$ of this distribution, in which case the weighting multiplies $g(y_{t+1} \mid x_{t+1})$ by the ratio $f(x_{t+1} \mid x_t)/q(x_{t+1} \mid y_{t+1}, x^{t+1})$. For the resampling condition, a common approach is based on an *effective sample size* (*ESS*) given by

$$ESS = \left(\sum_{j=1}^{N} (1/w_{t+1,j})^2 \right)^{1/2}, \tag{5}$$

where resampling occurs if $ESS < K$ for some K such that $1 < K < N$ (e.g., $K = N/2$).

The particle filtering method is an example of *sequential importance sampling*, a sequential implementation of the importance sampling technique for reducing variance in Monte Carlo estimation (see, e.g., Asmussen and Glynn [5]) such as evaluating an integral,

$$I(\phi, w) = \int_X \phi(x) w(x)\, dx, \tag{6}$$

where p is a probability density over x and ϕ is some performance function. A crude Monte Carlo estimate of $I(\phi, w)$ is given by

$$\hat{I}^{mc}(\phi, w) = \sum_{i=1}^{N} \phi(x_i)/N, \tag{7}$$

where the x_i are independent draws from the distribution corresponding to w. Importance sampling attempts to reduce the variance of $\hat{I}^{mc}(\phi, w)$ by finding a function q that approximates ϕ, where $q(x) > 0$ whenever $w(x) > 0$, and interpreting $I(\phi, w)$ as

$$I(\phi, w) = \int_X \frac{I(q, w)\phi(x)}{q(x)} \frac{w(x)q(x)}{I(q, w)}\, dx, \tag{8}$$

where $I(q, w) = \int_X q(x)w(x)\, dx$ so that $I(\phi, w) = I((I(q, w)\phi)/q, (w \cdot q)/I(q, w))$. Importance sampling then applies crude Monte Carlo simulation to $I((I(q, w)\phi)/q, (w \cdot q)/I(q, w)) = I((I(q, w)\phi)/q, \pi)$, where $\pi(x) = (w(x)q(x))/I(q, w)$, to obtain an estimate,

$$\hat{I}^{is-\pi}(\phi, w) = \sum_{i=1}^{N} \frac{I(q, w)\phi(x_i)}{q(x_i)} \Big/ N, \tag{9}$$

which then has smaller variance than $\hat{I}^{c}(\phi, w)$ if q is close to ϕ (and zero variance if $q = \phi$). This process, however, requires finding $I(q, w)$, which could be as difficult as finding $I(\phi, w)$. If q is tuned for a single ϕ, this is also problematic in sequential implementations where the target ϕ may be changing. An alternative is to assume q is a given density.

To remedy the problem of finding $I(q, w)$, this can be replaced with $\hat{I}^{mc}(q, w)$ to obtain a biased (but consistent) estimate,

$$\hat{I}^{is'-\pi}(\phi, w) = \sum_{i=1}^{N} \frac{\hat{I}^{mc}(q, w)\phi(x_i)}{q(x_i)} \Big/ N. \tag{10}$$

This provides an implementable scheme even when $I(q, w)$ is not available, because the random draws of x_i from a distribution proportional to $w \cdot q$ can be obtained with Gibbs sampling or, more generally, a Markov chain Monte Carlo method (following, e.g., Metropolis et al. [19]) without knowledge of $I(q, w)$.

The principle idea of the particle filter is to take advantage of this importance sampling scheme and to apply it sequentially across time to estimate the distribution of the unobserved quantities conditional on noisy observations. The typical estimation situation occurs when π corresponds to $p(x \mid y)$, $w \cdot q$ corresponds to $p(x, y)$ (i.e., w corresponds to $p(x)$

and q to $p(y \mid x)$), and $I(q,w)$ corresponds to $p(y)$. The objective is then to find $I(\phi,\pi) = I(\phi(\pi/q),q) = I(\phi(w/I(w,q)),q)$ using

$$\hat{I}^{is-q}(\phi,\pi) = \sum_{i=1}^{N} \frac{\phi(x_i)w(x_i)}{\hat{I}^{mc}(w,q)} \Big/ N, \tag{11}$$

where w may also be given as π/q. In this way, the observations x_i drawn with the density q are weighted by

$$\hat{w}_i = \frac{w(x_i)}{\sum_{i=1}^{N} w(x_i)}, \tag{12}$$

giving

$$\hat{I}^{is-q}(\phi,\pi) = \sum_{i=1}^{N} \phi(x_i)\hat{w}_i. \tag{13}$$

Particle methods employ this importance sampling procedure sequentially at each stage t, so that

$$q_t(x^t) = q_{t-1}(x^{t-1})q_t(x_t \mid x^{t-1}) \tag{14}$$

$$= q_1(x_1)\prod_{s=2}^{t} q_s(x_s \mid x^{s-1}), \tag{15}$$

$$w_t(x^t) = \frac{\pi_t(x^t)}{q_t(x^t)} = w_{t-1}(x^{t-1})\frac{\pi_t(x^t)}{\pi_{t-1}(x^{t-1})q_t(x_t \mid x^{t-1})} \tag{16}$$

$$= \frac{\pi_1(x_1)}{q_1(x_1)}\prod_{s=2}^{t} \frac{\pi_s(x^s)}{\pi_{s-1}(x^{s-1})q_s(x_s \mid x^{s-1})}, \tag{17}$$

which then provides the sequential updating procedure. Knowledge of $q_t(x_t \mid x^{t-1})$ then enables consistent estimates of π_t and integrals with respect to criterion ϕ. The SIR method can then be seen to follow this sequential importance sampling method, where $q_t(x_t \mid x^{t-1}) = f(x_t \mid x^{t-1})$ for all t and π_t corresponds to $p(x^t,y^t)$ so that

$$\frac{\pi_t(x^t)}{\pi_{t-1}(x^{t-1})q_t(x_t \mid x^{t-1})} \to \frac{p(x^t,y^t)}{p(x^{t-1},y^{t-1})f(x_t \mid x^{t-1})} = g(y_t \mid x_t). \tag{18}$$

Then the update of w_t corresponding to (16) and (12) is

$$\hat{w}_{t+1,j} = \frac{w_{t,j}g(y_{t+1} \mid x_{t+1})}{\sum_{j=1}^{N} \hat{w}_{t,j}g(y_{t+1} \mid x_{t+1})}, \tag{19}$$

as given in the SIR method above.

3. An Inventory Example

As an example of how the method applies to sequential estimation, consider the common operational problem of estimating inventory with imperfect observations (see DeHoratius and Raman [9], DeHoratius et al. [10]). In this case, the imperfect inventory observations are given by y_t. The goal is then to estimate the distribution of the actual inventory state x_t (and to compute various performance measures such as the expected level and probability of being stocked out). To use the SIR particle filter as given above, we suppose that the transitions from one period to the next, $f(x_t \mid x_{t-1})$ and the conditional distribution of the observation noise $g(y_t \mid x_t)$ are known.

To enable analytical comparisons, we assume the following linear dynamics and Gaussian disturbances so that the process fits the linear, quadratic Gaussian (LQG) format:

$$x_t = Fx_{t-1} + u_t, u_t \sim \mathbb{N}(0, Q), \tag{20}$$
$$y_t = Hx_t + v_t, v_t \sim \mathbb{N}(0, R), \tag{21}$$

where F and H are transition matrices, and $\mathbb{N}(\mu, \Sigma)$ refers to a Gaussian (normal) distribution with mean μ and covariance matrix Σ. With this specification, the Kalman [15] filter gives

$$x_t \mid y^t \sim \mathbb{N}(x_t'', P_t''), \tag{22}$$

where x_t'' and P_t'' are defined sequentially by the following steps given x_{t-1}'' and P_{t-1}'':

$$x_t' = Fx_{t-1}'', \tag{23}$$
$$P_t' = FP_{t-1}''F^T + Q, \tag{24}$$
$$z_t = y_t - Hx_t', \tag{25}$$
$$S = HP_t'H^T + R, \tag{26}$$
$$K = P_t'H^TS^{-1}, \tag{27}$$
$$x_t'' = x_t' + Kw_t, \tag{28}$$
$$P_t'' = (I - KH)P_t'. \tag{29}$$

To use the SIR particle filter method, we need the transition density $f(x_t \mid x_{t-1})$, which is given by the normal density with mean Fx_{t-1} and covariance R, and the conditional observation density $g(y_t \mid x_t)$, which is the normal density with mean Hx_t and covariance Q. The Kalman filter provides the desired $p(x_t \mid y^t)$ in the pure LQG case, but the particle filter method applies generally and, therefore, can handle realistic situations such as truncation at zero for stockouts or at some upper limit of stocking capacity.

The results of a single simulation path are given in Figure 1, which shows the actual x_t (designated by asterisks) and observed y_t (designated by plus signs) values for 10 periods. The Kalman filter distribution estimate of x_{10} given y^{10} is $\mathbb{N}(13.8, 0.62)$. We compare this distribution to the result of the SIR using $N = 50$ particles and resampling when $ESS < 25$ in

FIGURE 1. Actual and observed value for 10 periods.

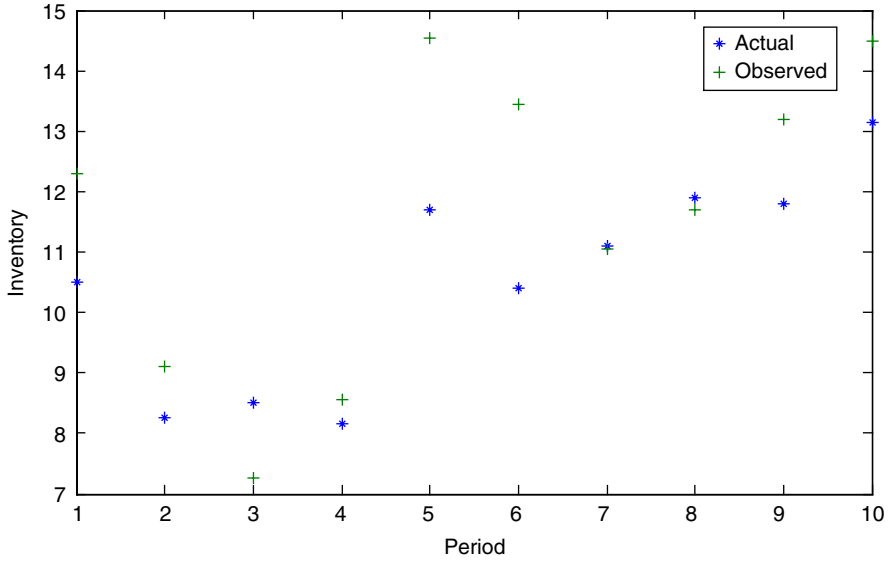
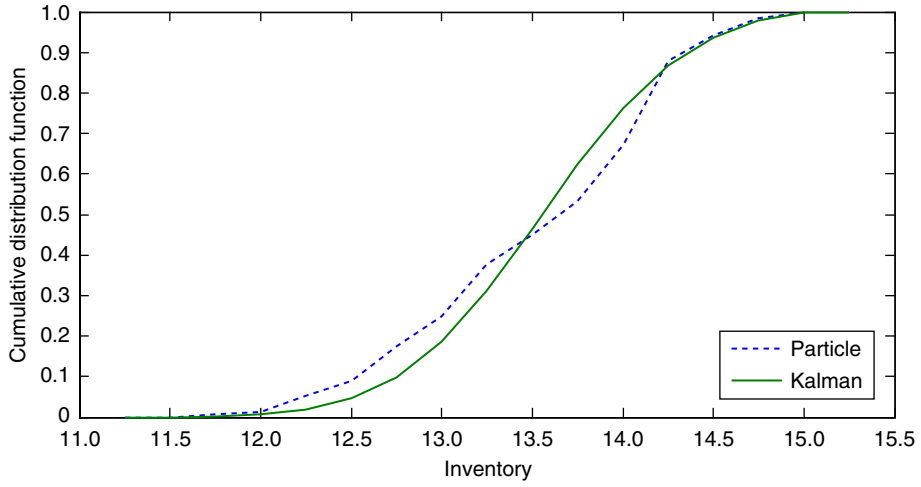


FIGURE 2. Kalman filter and SIR particle filter cumulative distribution functions.

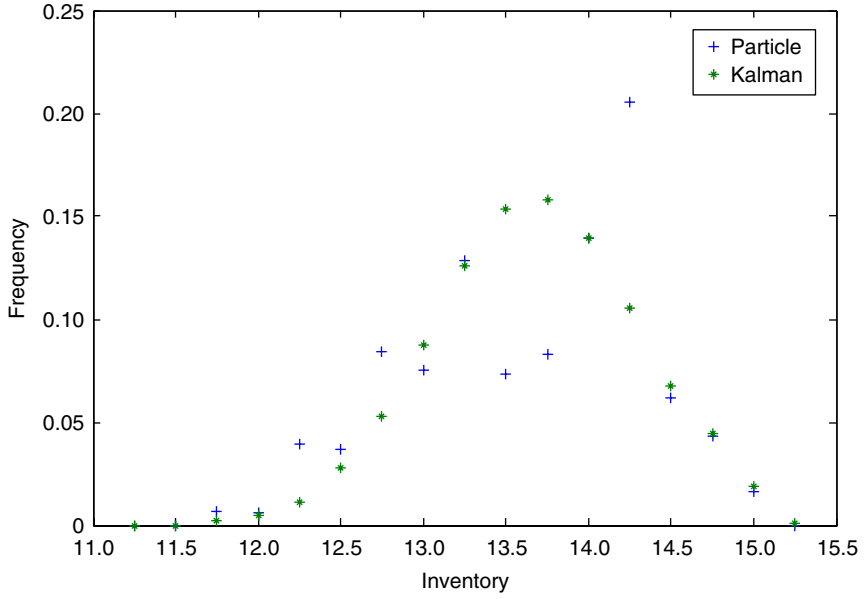


Figures 2 and 3. Figure 2 shows the cumulative distribution function from both the Kalman filter (exact) and the SIR estimate, whereas Figure 3 provides histograms to show the relative frequency of each with intervals of length 0.25. The plots of the distributions show that the particle filter, which only assumes the one-step transition distribution and a distribution on observation noise, provides a reasonably accurate representation of the true distribution given by the Kalman filter. Of course, in this simple case, when the distribution is available explicitly, the particle filter provides little advantage, but in other cases where computing the true distribution is intractable, the particle filter provides an effective alternative.

4. Uses for Simulation

The central ideas in particle filtering are the uses of the importance step in reweighting each new particle generated sequentially in time using the weight of the next observation

FIGURE 3. Kalman filter and SIR particle filter frequency comparisons.



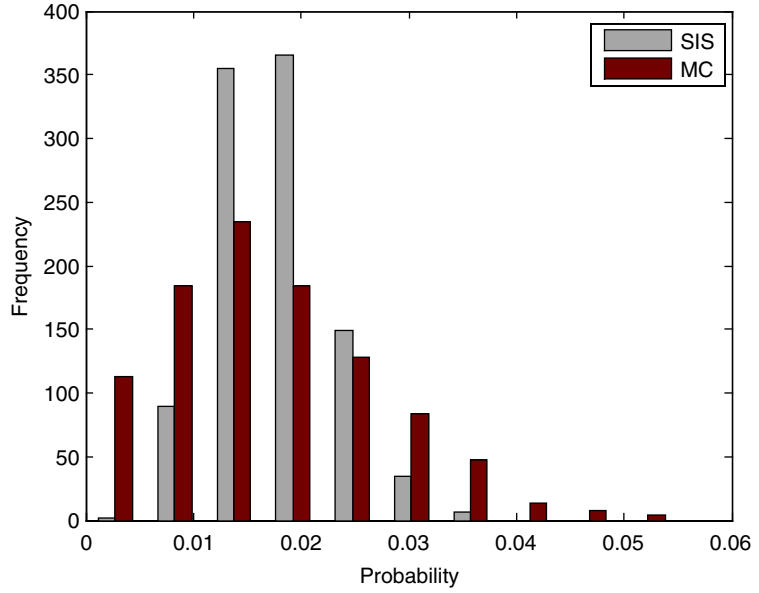
distribution conditional on that particle and resampling when the weights on the branches become too imbalanced. This general approach can be applied in many different simulation settings. As the example in the previous section suggests, one possibility is to use this approach in simulating possible sample paths that would be consistent with a set of observations.

This sequential version of importance sampling can be particularly useful for situations, such as risk management, where the central interest is in tail behavior (e.g., where ϕ is an indicator of loss at a given level). Estimation of such risk measures suffers from high variance because tail events are infrequently observed in standard Monte Carlo implementations. Using SIS as in the particle filter method, samples of x_t can be drawn from $q_t(x_t \mid x_{t-1})$, which focuses the samples on the area of interest, such as low x_t values of a particular asset or portfolio. The result is that many more samples occur in the tails of the overall distribution of x_t, lowering the variance in risk measure estimates.

As an example, Figure 4 compares the results of using a particle or sequential importance sampling method to those using crude Monte Carlo simulation for estimating the probability of a 50% loss after five periods for a portfolio with a normally distributed mean log return of 5% and standard deviation of log return of 20% in each period (i.e., where returns are log-normal with these parameters in each period). The figure shows the histograms of the fraction of those samples with losses greater than 50% over five periods for 1,000 replications of simulations with 200 particles or sample paths in each replication. For the importance distribution q_t, the particle method also uses a log-normal return, but with zero mean log return to more closely match the target distribution over the low-return range. For these replications, the particle method replications have a mean of 1.76% (compared to a true value of 1.75%) with a standard deviation of 0.52% (giving a standard error 0.0164%), whereas the crude Monte Carlo sample probabilities have a mean of 1.80% with a standard deviation of 0.93%. In this case, sequential importance sampling reduces the sampling standard error by almost one-half. With other choices of the importance distribution tuned to criteria of interest, much greater error reductions can be possible.

The particle filter method also relates to the *importance splitting* method (see, e.g., L'Ecuyer et al. [18], Glasserman et al. [12], Lagnoux [17], Morio et al. [20]) to obtain the

FIGURE 4. Histograms of particle method (SIS) and crude Monte Carlo (MC) results for probability of 50% portfolio loss.



probability ($p(A)$) of a rare event A by finding $p(A) = p(A_0)\prod_{i=1}^{k} P(A_i \mid A_{i-1})$, where $A_0 \supset A_1 \supset \cdots \supset A_k = A$ using estimates of each $P(A_i \mid A_{i-1})$. This can be seen as a version of particle filtering where g corresponds to an indicator of membership in A_i. This then allows estimates of the conditional survival rates using $\sum_{j=1}^{N} \hat{w}_{t,j} g(y_{t+1} \mid x_{t+1})$ (see Amrein and Künsch [2]).

5. Uses in Learning and Optimization

The particle filter method also applies to problems in which some of the dynamics are unknown and must also be inferred from the observations. These approaches are known as *particle learning* (see Polson et al. [23], Carvalho et al. [8]). In this case, we augment the full state to $z_t = (x_t, s_t, \theta)$, where s_t is a sufficient statistic for the underlying parameters θ. The method then follows the procedure below (from Carvalho et al. [8]).

Particle learning method.

(1) *Resample.* Draw N samples $z'_{t,j}$, $j = 1, \ldots, N$ from $z_{t,j} = (x_{t,j}, s_{t,j}, \theta_j)$ with weights $\hat{w}_{t,j} \propto p(y_{t+1} \mid z_{t,j})$.

(2) *Propagate.* Draw $x_{t+1,j}$, $j = 1, \ldots, N$ using $p(x_{t+1} \mid z'_{t,j}, y_{t+1})$.

(3) *Propagate.* Draw $s_{t+1,j}$, $j = 1, \ldots, N$ using $p(s_{t+1} \mid x'_{t+1}, s'_t, y_{t+1})$.

(4) *Sample.* Draw θ_j, $j = 1, \ldots, N$ from $p(\theta \mid s_{t+1})$.

The particle learning method then yields estimates of both the parameters and the states x_t. By examining the conditional mode of the distribution over θ, the process identifies the most likely distribution consistent with the observations. Identifying the mode then suggests an alternative for optimization by identifying a distribution with a density proportional to an objective function. Sampling from this distribution then provides a Monte Carlo optimization procedure as in simulated annealing (see, e.g., Kirkpatrick et al. [16], Aarts and Korst [1]). This approach is described in Pincus [21, 22], where minimizing a function $\phi(x)$ over a set X with a unique optimum is obtained by sampling from the distribution $p(x, \kappa)$ given by

$$p(x, \kappa) \propto e^{-\kappa \phi(x)}, \tag{30}$$

where increasing κ concentrates the distribution around the mode (which also coincides asymptotically with the mean). This approach can be applied with the particle filtering apparatus by interpreting $f(x_t \mid x_{t-1})$ as any distribution that covers X and interpreting $g(y \mid x)$ as proportional to $e^{-\kappa\phi(x)}$. Various alternatives can also be designed with f also including objective information and κ being chosen adaptively.

As an example of this approach, consider finding the minimum value of the Rosenbrock [24] function, given by

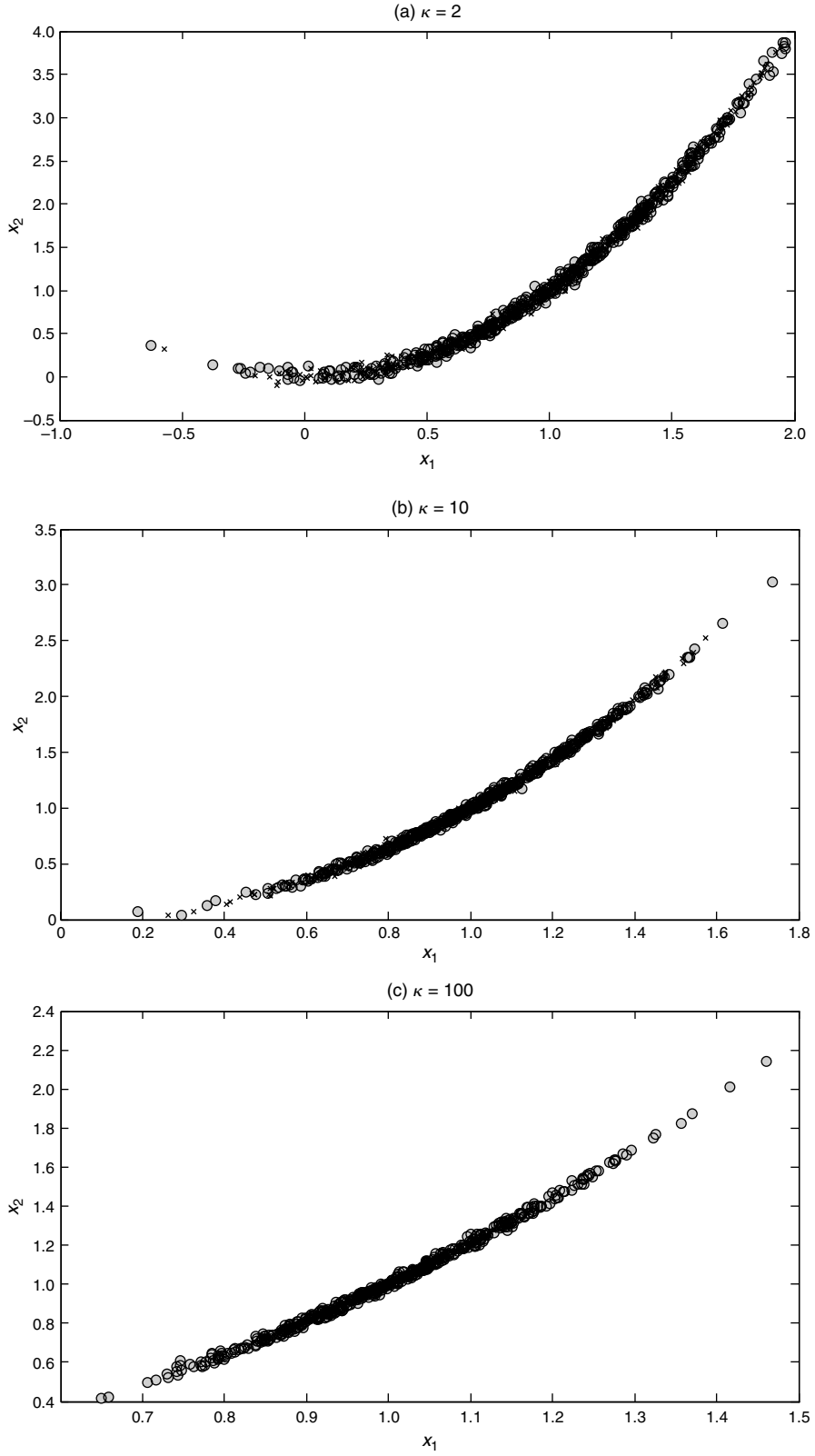
$$\phi(x = (x_1, x_2)) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2, \tag{31}$$

which has a global minimum at $x^* = (1, 1)$, but which is notoriously difficult to obtain with standard local search procedures. For this implementation, we use a specification for f from the following updating method:

$$x_t = \begin{cases} u & \text{if } v \le e^{-\kappa(\phi(u)/\phi(x_{t-1}))}, \\ x_{t-1} & \text{otherwise,} \end{cases} \tag{32}$$

where $u \sim \mathrm{U}([-4, 4]^2)$, a uniform distribution over $[-4, 4] \times [-4, 4]$, and $v \sim \mathrm{U}([0, 1])$, a standard uniform draw. This approach corresponds to the Metropolis–Hastings method of Markov chain Monte Carlo estimation (see Metropolis et al. [19], Hastings [14]) using the distribution proportional to $e^{-\kappa\phi(x)}$ for the acceptance criterion. In this case, we use the same κ as used in g (although in general they may be different). Figure 5, (a)–(c), shows the results of using this method without resampling (the pure Metropolis–Hastings method is indicated by circles) for $T = 5{,}000$ iterations with $N = 500$ sample runs and for the same T and N

FIGURE 5. Results from pure Metropolis–Hastings (circles) and SIR methods (crosses) for the Rosenbrock [24] function optimization with objective parameters, $\kappa = 2, 10$, and 100.



with resampling (the SIR particle method indicated by crosses) whenever $ESS < 250$. The figure shows how the results cluster more tightly around the optimal solution $x^* = (1, 1)$ for larger κ values (noting that the scales change for the higher κ values). The advantage of resampling is shown by the lower dispersion of the crosses compared to the circles. This is especially prominent for the high κ values, where the results are most precise and it is most valuable to avoid sample paths that have low objective weight.

This approach can be expanded further for dynamic optimization in which controls u_t are chosen conditional on the state variables x_t. The process involves stepping forward to obtain a new random component observation y_t conditional on x_t and u_t and resampling u_t conditionally on that observation. This is applied in Birge and Polson [6], where it is shown that this process provides unbiased estimates of optimal LQG controls and can be applied consistently in more general dynamic contexts.

6. Conclusions

This tutorial has introduced the basic concepts in particle filtering methods, particularly the ideas of sequential importance sampling and resampling, and their potential applications in the operations research domains of Monte Carlo simulation and optimization. Examples were given to illustrate how the methods work and their relative advantages in reducing the variance of performance criteria and optimal solution estimates. As noted in the introduction, many other tutorials and the papers cited throughout give further background on the theory underlying these methods and their applications for statistical estimation.

Acknowledgments

The author is grateful for comments on this work from workshop participants at Cornell University and at the 2011 INFORMS Midwest Regional Conference, Ohio State University, Columbus, Ohio. This work was supported by the University of Chicago Booth School of Business.

References

[1] E. Aarts and J. Korst. *Simulated Annealing and Boltzmann Machines*. John Wiley & Sons, New York, 1989.

[2] M. Amrein and H. R. Künsch. Importance splitting for rare event estimation. Presentation, University of North Carolina at Chapel Hill, Chapel Hill, February 22, http://stat.ethz.ch/~kuensch/talks/chapelhill10.pdf, 2010.

[3] C. Andrieu and A. Doucet. Particle filtering for partially observed Gaussian state space models. *Journal of the Royal Statistical Society: Series B* 64(4):827–836, 2002.

[4] S. Arulampalam, S. Maskel, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing* 50(2):174–188, 2002.

[5] S. Asmussen and P. Glynn. *Stochastic Simulation*. Springer-Verlag, New York, 2008.

[6] J. R. Birge and N. G. Polson. Multi-stage dynamic programming via sequential sampling. Working paper, The University of Chicago Booth School of Business, Chicago, 2012.

[7] O. Cappé, S. J. Godsill, and E. Moulines. An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE* 95(5):899–924, 2007.

[8] C. M. Carvalho, M. S. Johannes, H. F. Lopes, and N. G. Polson. Particle learning and smoothing. *Statistical Science* 25(1):88–106, 2010.

[9] N. DeHoratius and A. Raman. Inventory record inaccuracy: An empirical analysis. *Management Science* 54(4):627–641, 2008.

[10] N. DeHoratius, A. J. Mersereau, and L. Schrage. Retail inventory management when records are inaccurate. *Manufacturing and Service Operations Management* 10(2):254–277, 2008.

[11] A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. D. Crisan and B. Rozovskiĭ, eds. *The Oxford Handbook of Nonlinear Filtering*. Oxford University Press, Oxford, UK, 656–704, 2011.

[12] P. Glasserman, P. Heidelberger, P. Shahabuddin, and T. Zajic. Splitting for rare event simulation: Analysis of simple cases. L. F. Perrone, B. Lawson, J. Liu, and F. P. Wieland, eds. *Proceedings of the Winter Simulation Conference WSC 2006, Monterey, CA*, 302–308, 2006.

[13] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEEE Proceedings F* 140(2):107–113, 1993.

[14] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57(1):97–109, 1970.

[15] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME Journal of Basic Engineering* 82:35–45, 1960.

[16] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science* 220(4598):671–680, 1983.

[17] A. Lagnoux. Rare event simulation. *Probability in the Engineering and Informational Sciences* 20(1):45–66, 2006.

[18] P. L'Ecuyer, V. Demers, and B. Tuffin. Splitting for rare event simulation. L. F. Perrone, B. Lawson, J. Liu, and F. P. Wieland, eds. *Proceedings of the Winter Simulation Conference WSC 2006, Monterey, CA*, 137–148, 2006.

[19] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics* 21(6):1087–1091, 1953.

[20] J. Morio, R. Pastel, and F. Le Gland. An overview of importance splitting for rare event simulation. *European Journal of Physics* 31(2):1295–1303, 2010.

[21] M. Pincus. A closed form solution of certain programming problems. *Operations Research* 16(3):690–694, 1968.

[22] M. Pincus. A Monte Carlo method for the approximate solution of certain types of constrained optimization problems. *Operations Research* 18(6):1225–1228, 1970.

[23] N. G. Polson, J. Stroud, and P. Müller. Practical filtering with sequential parameter learning. *Journal of the Royal Statistical Society: Series B* 70(2):413–428, 2008.

[24] H. H. Rosenbrock. An automatic method for finding the greatest or least value of a function. *Computer Journal* 3(3):175–184, 1960.