02427 Advanced Time Series Analysis
Date: 04 January, 2019

Maximization of Sequential Monte Carlo (SIR) likelihood function estimates using Simultaneous Perturbation Stochastic Approximation

M. Wamberg [s146791][1] & B. A. Hastrup [s151602][1]

Abstract
In this paper we address the challenge of maximizing the noisy likelihood function obtained from Sequential Monte Carlo algorithms (SMC), also known as particle filters. A particular emphasis will be placed on the problem of estimating unknown parameters in non-linear and non-Gaussian state-space models. As a solution for effectively optimizing the SMC methods we consider Simultaneous Perturbation Stochastic Approximation (SPSA), which is a technique to approximate the gradient via a randomized finite difference method.

Keywords
Particle filters — Sequential Monte Carlo — MLE — SA — SPSA — Stochastic optimization

[1] *DTU Compute, Technical University of Denmark, Denmark*

Introduction

SMC methods are a class of Monte Carlo algorithms used to estimate the internal states in a dynamical system based on some noisy observations arriving sequentially in time. This is called online estimation.
In general, one seeks to determine and update the posterior distributions of the states in a Markov process as the noisy observations become available. Examples in which this is applicable are numerous, among others time series analysis, signal and image procession, target tracking etc. Common to these applications is that it is more advantageous to update the previously determined posterior distribution than to recalculate them from scratch.

The SMC methods have the great advantage of not imposing linearity requirements on the state-space model considered. This is also the case for the initial state and the noise additions, they are not subject to any specific requirements and therefore can take any particular form. Thus, the SMC algorithms constitute a very suitable method for obtaining samples from a desired distribution without having to make restrictive assumptions about either the state-space model or the state distributions.
However, there are also some shortcomings in the standard SMC methods. One of the most severe limitations is that the standard likelihood approximation is non-smooth in the parameter space within the SMC framework, which makes it challenging to obtain maximum likelihood estimates of the parameters.
In this paper, we focus on solving this challenging problem by using the SPSA method introduced by Spall [1]. Optimization via SPSA is done through a random search in the parameter space and requires only two measurements from the objective function regardless of the parameter dimension. The method provides several desirable features, obviously the method is suitable for high-dimensional problems due to the efficient gradient approximation using only two measurements, but it also allows for the input data to include added noise, which makes the method particularly suitable for Monte Carlo simulations.

1. State space models and SMC methods

Consider a non-linear and non-Gaussian state model. We denote the unobserved sequence of states $\{x_t\}_{t=0}^{\infty}$, with $x_t \in \mathscr{X}$ and initial distribution $p(x_0)$. Then $p(x_t|x_{t-1})$ is the transition probability for $t \geq 1$. Let $\{y_t\}_{t=1}^{\infty}$ denote the observations, with $y_t \in \mathscr{Y}$. Given $\{x_t\}$, the observations are assumed to be conditional independent and we denote $p(y_t|x_t)$ as our observation probability for $t \geq 0$.
We are interested in estimating the posterior distributions of the unobserved states $p(x_{0:t}|y_{1:t})$ recursively and the expectations under these posteriors

$$\mathbb{E}_{p(x_{0:t}|y_{1:t})}[f_t(x_{0:t})] = \int f_t(x_{0_t}) p(x_{0:t}|y_{1:t}) dx_{0:t}, \tag{1}$$

for some function $f_t : \mathscr{X}^t \to \mathbb{R}^n$ integrable with respect to $p(x_{0:t}|y_{1:t})$.
Through the use of Bayes' theorem we can obtain the following expression for the posterior distribution at any time

$$p(x_{0:t}|y_{1:t}) = \frac{p(y_{1:t}|x_{0:t})p(x_{0:t})}{\int p(y_{1:t}|x_{0:t})p(x_{0:t})dx_{0:t}}. \tag{2}$$

Using the definition of conditional densities and the assumption regarding conditional independent observations together with the Markov property contained by $\{x_t\}$ we are able to obtain the following recursion [2]

$$p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1}, \quad (3)$$

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{\int p(y_t|x_t)p(x_t|y_{1:t-1})dx_t}. \quad (4)$$

To solve this recursion, it is necessary to make numerical approximations. Since it is typically not possible to draw samples from the posterior distribution a *Sequential importance sampling* (SIS) can be used. In the SIS, the samples are drawn from an importance sampling distribution factored as

$$\pi(x_{0:t}|y_{1:t}) = \pi(x_0)\Pi_{k=1}^{t}\pi(x_k|x_{0:k-1}, y_{1:k}). \quad (5)$$

Each draw has a corresponding weighting, which in SIS can be evaluated recursively by [2]

$$w(x_{0:t}) = \frac{p(x_{0:t}|y_{1:t})}{\pi(x_{0:t}|y_{1:t})}, \quad (6)$$

which results in the normalized importance weights

$$\tilde{w}_t^{(i)} = \frac{w(x_{0:t}^{(i)})}{\sum_{j=1}^{N} w(x_{0:t}^{(j)})}. \quad (7)$$

As $t \to \infty$, there will occur skewness in the distribution, since the importance weights for many of the particles will approach zero. This phenomenon is termed *particle degeneracy*, and it prevents good likelihood approximations because the calculations of the underlying integrals are simply performed by using only a few particles with non-zero weight, resulting in a large variance. Thus, in order to avoid skewness in the distribution of the importance weights $\tilde{w}_t^{(i)}$ a resampling step is introduced. The general idea with resampling is to eliminate particles with low importance weight and multiplying particles with high importance weights. This ensures that propagation occurs in the areas where there is most particle mass. The resampling is performed by sampling N times with replacement from the weighted set in order to generate a new sample of N particles. Since the resampling is carried out with replacement, there is a high probability that particles with high importance weight are drawn many times, while particles with low importance weight are probably not drawn at all. Next, we replace the old particles with the new ones $\{\tilde{x}\}_{i=1}^{N}$, that is copies of the particles that are drawn. Once the resampling step is complete all the weights are reset to $w_t^{(t)} = 1/N$. From here we propagate the resampled particles to $t+1$, so we create a new set of particles using only the particles that are resampled. Thus, for each time step there will still be N particles, despite some particles being extinct in the previous step, since those particles with high importance weights have been copied.

1.1 Bootstrap filter

The so-called *bootstrap filter* is an algorithm that implements the resampling procedure in the sequential importance sampling. In the bootstrap filter it is assumed

$$\pi(x_t^{(i)}|x_{0:t-1}^{(i)}, y_{1:t}) = p(x_t^{(i)}|x_{t-1}^{(i)}), \quad (8)$$

i.e. the law of the Markov process forms the basis for the importance sampling, which enables us to simplify the weights

$$w_t^{(i)} = N^{-1}p(y_t|x_t^{(i)}). \quad (9)$$

The full bootstrap filter implementation algorithm follows below, where the output is the likelihood function for a given θ parameter (see section 2)

Bootstrap filter algorithm

```
# initialization at t=0
for $i = 1:N$
  Sample $x_0^{(i)} \sim p(x_0|\theta)$
  Assign $\tilde{w}_0^{(i)} = N^{-1}$

for $t = 1:T$
  # importance sampling
  for $i = 1:N$
    Sample $\tilde{x}_t^{(i)} \sim p(x_t^{(i)}|x_{t-1}^{(i)}, \theta)$
    Compute $w_t^{(i)} = p(y_t|\tilde{x}_t^{(i)}, \theta)$
    Normalize $\tilde{w}_t^{(i)} = w_t^{(i)} / \sum w_t^{(i)}$
  # resampling
  Sample $N$ particles $\{x_{0:t}^{(i)}\}$ with replacement from
    $\{\tilde{x}_{0:t}^{(i)}\}$ according to $\{\tilde{w}_t^{(i)}\}$
return $\hat{p}(y_{1:T}|\theta) = \Pi_{t=1}^{T}\left(N^{-1}\sum_{i=1}^{N} w_t^{(i)}\right)$
```

2. Parameter estimation

Now consider a state space model with its latent variable transition probability dependent on a parameter ξ and its observation probability dependent on a parameter η, i.e. $p(x_t|x_{t-1}, \xi)$ and $p(y_t|x_t, \eta)$ denotes these probabilities. Both parameters are unknown static parameters and we assume that the considered system evolves according to the combined parameter $\theta^* = (\xi^*, \eta^*)$. Then we wish to determine this parameter θ^* which is particularly a challenging task for a non-linear and non-Gaussian system. A standard method for doing this is to maximize the log-likelihood function. Using the definition of the likelihood function as being a function of the parameter θ equal to the density of the observed data y we have

$$\begin{aligned} L(\theta|y_{1:T}) &= p(y_{1:T}|\theta) = \Pi_{t=1}^{T}p(y_t|y_{1:t-1}, \theta) \\ &= \Pi_{t=1}^{T}\int p(y_t|x_t, \eta)p(x_t|y_{1:t-1}, \theta)dx_t \\ &= \Pi_{t=1}^{T}\mathbb{E}[p(y_t|X_t, \eta)|y_{1:t_1}, \theta]. \end{aligned} \quad (10)$$

Thus, in order to approximate the expression in equation (10) it is necessary to obtain the filter samples $(x_{pr,t}^{(j)})$ which are disturbed according to $p(x_t|y_{1:t-1},\theta)dx_t$. We can then approximate the log-likelihood by taking the logarithm on both sides of equation (10) and get

$$\log L(\theta|y_{1:T}) = -T\log N + \sum_{t=1}^{T} \log\left(\sum_{j=1}^{N} p(y_t|x_{pr,t}^{(j)},\theta)\right). \quad (11)$$

The log-likelihood function that we obtain through this formula has two main disadvantages, the first being that in order to obtain the likelihood for different θ values we must generate new filter samples each time, the other is that the error for these different θ values are independent resulting in a noisy likelihood function. We are therefore looking toward SPSA to maximize the log-likelihood function. Figure 1 illustrates the MLE challenge for Monte Carlo estimates.

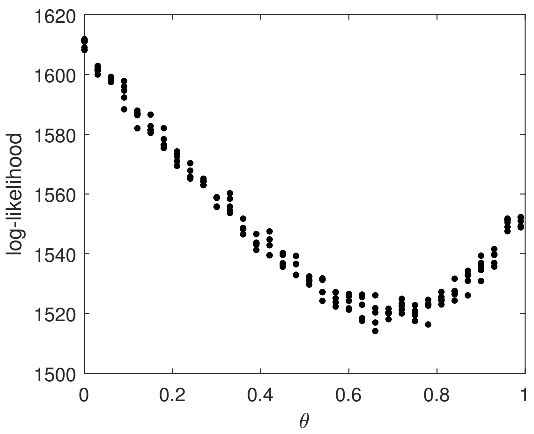

Figure 1. Particle filter estimates of the log-likelihood function when only the AR(1)-coefficient, θ, is unknown (see Section 4 for the full model where also the variances are estimated). The true θ-value is 0.7, and here 5 particle filter estimates, with N=150 particles in each filter, are calculated for each trial value of θ. (Obviously, we will calculate the log-likelihood at each θ as the mean of the five log-likelihoods, or whatever number of independent filters we use.)

3. Optimizing SMC by using SPSA

The problem of minimizing a differentiable loss function $L(\theta)$ can be described by finding the zeros of the gradient $\nabla L(\theta)$. That is, we want to find the minimizing θ^* such that $\nabla L(\theta) = \partial L/\partial\theta = 0$, we can write the procedure to estimate this θ^* recursively as

$$\hat{\theta}_{t+1} = \hat{\theta}_t - a_t\hat{g}_t(\hat{\theta}_t), \quad (12)$$

where $\hat{g}_t(\hat{\theta}_t)$ is the noisy estimate of the gradient $\nabla L(\theta)$ estimated at $\hat{\theta}_t$, and a_t is a nonnegative gain sequence satisfying $a_t \to 0$ and $\sum_{t=1}^{\infty} a_t = \infty$. Under the right conditions the iteration given in equation (12) will converge to θ^*.

In order to obtain the gradient estimate $\hat{g}_t(\hat{\theta}_t)$ we introduce SPSA, which makes use of the finite difference method for its approximation of the gradient estimate. Denoting a measurement by $y(\cdot)$, the SPSA method requires two measurements of the form $y(\hat{\theta}_t \pm \text{perturbation})$. The perturbation is performed simultaneously on all elements using a random perturbation vector Δ_t. Then, the estimated gradient elements can be written

$$\hat{g}_{ti}(\hat{\theta}) = \frac{y(\hat{\theta}_t + c_t\Delta_t) - y(\hat{\theta}_t - c_t\Delta_t)}{2c_t\Delta_{ti}}, \quad (13)$$

where $\Delta_k = (\Delta_{t,1},\ldots,\Delta_{t,p})^T$ is a $p-$dimensional random perturbation vector, c_t denotes a sequence of positive numbers such that $c_t \to 0$. Generally, the two sequences a_t and c_t take the form $a_t = a/(t+A)^{\alpha}$ and $c_t = c/t^{\gamma}$, where a,c,A,α and γ are non-negative coefficients to which initial values are to be guessed.

Convergence is almost certainly achieved in equation (12) if $L(\theta)$ is sufficiently smooth near θ^*. Furthermore, the $\{\Delta_{ti}\}$ are required to be mutually independent random variables, symmetrically distributed around zero and with finite inverse moments $E(|\Delta_{ti}|^{-1})$ which rules out the possibility of division by zero [3]. A distribution that meets these requirements is the Bernoulli ±1 distribution, which is therefore a popular choice for Δ_{ti}. Usually, different constants a,c,A,α and γ will be specified for each dimension of the parameter vector, θ, especially when the magnitudes of the parameters differ and when the loss function has different sensitivities w.r.t. to these parameters. In that case, c_t is a $p \times p$-matrix. Below is summarized the SPSA in algorithm form.

SPSA optimization algorithm

```
# initialization
Pick initial guess $\hat{\theta}_0$ and coefficients $a,c,A,\gamma$

for t=1:T
  Assign $\Delta_t \leftarrow \text{Bernoulli}(0.5)$

# gradient approximation
Compute $y_t^+ = y(\hat{\theta}_t + c_t\Delta_t)$
Compute $y_t^- = y(\hat{\theta}_t - c_t\Delta_t)$

for t=1:T
  Compute $\hat{g}_t = (y_t^+ - y_t^-)/(2c_t\Delta_t)$
  # update estimate
  Evaluate $\hat{\theta}_{t+1} = \hat{\theta}_t - a_t\hat{g}_t$
terminate if little change in successive iterates or
maximum number of iterations is reached.
```

It is worth noting that in our case where we want to maximize our log-likelihood function via the above SPSA algorithm that the $y(\cdot)$ function simply corresponds to the log-likelihood function defined in equation (11).

Figure 2 illustrates the SPSA minimization of some deterministic function (without noise) given by

$$f(x_1,x_2) = 11 - 2x_1 + 22x_1^2 - 20x_2 + 10x_2^2 + |x_1| + |x_2|, \quad (14)$$

for two different distributions from which Δ_{ti} is sampled. As the bishop on the chess board, the SPSA parameter path is confined to these diagonal pieces, when the Bernoulli distribution is used with identical constants a, c, A, α and γ for both dimensions (black graph). If we instead sample the Δ_{ti}'s from a uniform distribution on the interval $[-1,-1/10]\bigcup[1/10,1]$, a more fuzzy path will be taken towards the optimum (red graph). No actual calibration is required for this function and the constants were set to $A = 1$, $\alpha = 0.00$, $a = 0.01$, $c = 0.01$, $\gamma = 0.00$.

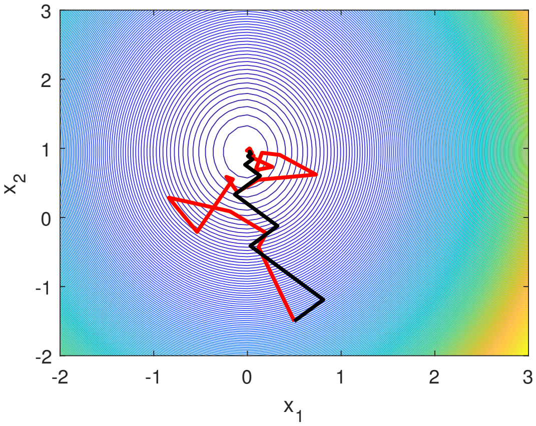

Figure 2. SPSA optimization of a noiseless loss function using a Bernoulli distribution (black) and a uniform distribution on $[-1,-1/10]\bigcup[1/10,1]$ (red) to sample the Δ_t-vector.

4. Applications

In this section we show how the SPSA method can be used to estimate the parameters of two different simulated toy models. First we consider the linear Gaussian state space model

$$x_n = \phi x_{n-1} + \sigma_v V_n, \quad x_0 \sim \mathcal{N}\left(0, \frac{\sigma_v^2}{1-\phi^2}\right) \tag{15}$$

$$y_n = x_n + \sigma_w W_n. \tag{16}$$

where $V_n \sim \mathcal{N}(0,1)$ and $W_n \sim \mathcal{N}(0,1)$ are both i.i.d. noise terms. Specifically, y_n is conditionally independent of any previous observations $y_{n' \le n}$ given x_n.

Figure 3 illustrates a successful parameter estimation, apart from a slight overestimation of the noise in the state evolution process (equation (15)).

In this specific estimation we chose a simple structure of the gain sequence a_n:

$$a_n = 0.99 a_{n-1}, \quad a_0 = 5 \cdot 10^{-4}, \tag{17}$$

while the other gain sequence

$$c_n = \frac{(0.01, 0.02, 0.025)^T}{n^{0.101}}. \tag{18}$$

Trial and error caused us to settle with these hyper parameters and no deeper quantitative/algorithmic analysis was used. For the next model different types of parameters will be estimated, which might influence the log-likelihood differently, so the hyper parameters will have to be more or less recalibrated entirely. But the take home message from this linear model estimation is that it proves advantageous to force a_n to zero much faster than c_n.

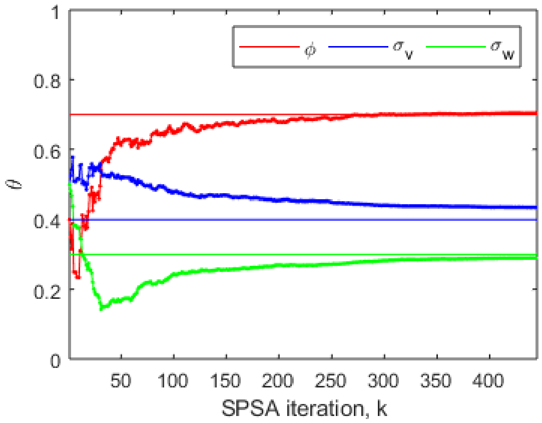

Figure 3. SPSA parameter convergence for a simulated data series $\{y_k\}$, $k = 1,..,200$ of the linear model from equation 15 and (16). At each log-likelihood measurement 10 independent particle filters are implemented with K=2000 particles in each. The true parameters are drawn as plain horizontal lines with $\theta^* = (0.7, 0.4, 0.3)^T$. The parameter estimate was initialized at $\theta_0 = [0.4, 0.5, 0.5]^T$.

Next, we examine a bimodal non-linear, non-Gaussian model. We use the standard dynamic model [4],[5]

$$x_n = \theta_1 x_{n-1} + \theta_2 \frac{x_{n-1}}{1+x_{n-1}^2} + \theta_3 \cos(1.2n) + \sigma_v V_n \tag{19}$$

$$y_n = c x_n^2 + \sigma_w W_n, \tag{20}$$

where $\sigma^2 = 10, c = 0.05, \sigma_w = 1, x_0 \sim \mathcal{N}(0,2), V_n \sim \mathcal{N}(0,1)$ and $W_n \sim \mathcal{N}(0,1)$. Thus, this example is non-linear both in the system and in the measurement equation. In this simulation, the true parameter is $\theta^* = (0.5, 25, 8)^T$ and the SPSA is initialized at $\theta_0 = (0.2, 20, 8)^T$. We see that θ_1 is estimated correctly while θ_2 and θ_3 are both off by one. Since, this is a nonlinear problem, and thereby harder to estimate, maybe we should have had more patience regarding convergence, i.e. have used smaller values for α which penalizes long run parameter step sizes in the gain sequence:

$$a_k = \frac{a}{(A+k)^\gamma}. \tag{21}$$

Here we used

$$a = [0.00005; 0.0006; 0.0003],$$

$$\alpha = [0.0008; 0.002; 0.007],$$

and $A = 10$. It is good to have a high value of A, since this stabilizes the parameter search in the first few iterations when the gradient approximations are most volatile. Otherwise parameters might escape their natural bounds and cause the particle filter to crash.

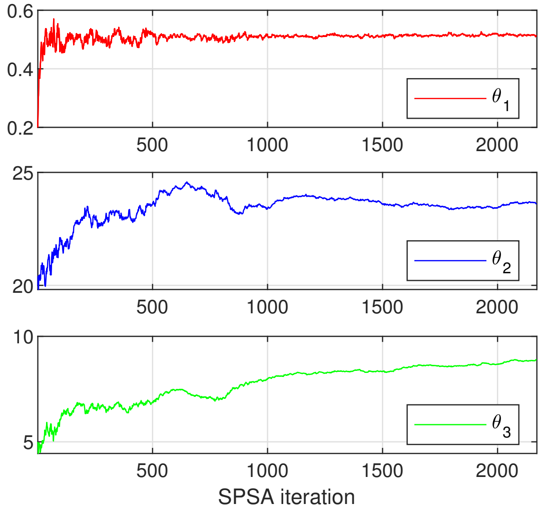

Figure 4. Convergence plot for the parameter estimates of the model in equation (19) and (20).

Finally, we seek to estimate the non-linear *stochastic volatility model*, which in the financial econometrics literature is used to model returns. The dynamics are given by

$$x_n = \phi x_{n-1} + \sigma V_n, \qquad X_0 \sim N\left(0, \frac{\sigma^2}{1-\phi^2}\right), \tag{22}$$

$$y_n = \beta \exp\left(\frac{x_n}{2}\right) W_n, \tag{23}$$

where $V_n \sim \mathcal{N}(0,1)$ and $W_n \sim \mathcal{N}(0,1)$ are both i.i.d. Here the observed process describes returns, while the latent process describes the evolution of the log-volatility of returns. The exponential transform is motivated by the desire to model the skewed distribution of squared returns that are observed empirically.

Unfortunately, the estimates did not converge. For some runs, 2 of the three parameters converged well. See code STOCHVOL and loss3 for implementation.

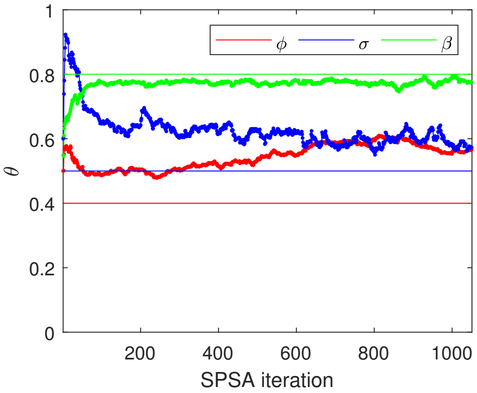

Figure 5. Convergence plot for the parameter estimates of the model in equation (22) and (22).

References

[1] Spall, J.C., `https://www.jhuapl.edu/sPsA/`

[2] Doucet, A., Freitas, N.D., Gordon, N., ”An Introduction to Sequential Monte Carlo Methods”, Sequential Monte Carlo Methods in Practice pp 3-14

[3] Spall, J.C., ”An Overview of the Simultaneous Perturbation Method for Efficient Optimization”, Johns Hopkins APL Technical Digest, vol. 19(4), pp. 482–492.

[4] Poyiadjis, G., Singh, S.S., and Doucet, A., ”Gradient-free Maximum Likelihood Parameter Estimation with Particle Filters”, Proceedings of the American Control Conference, 14-16 June 2006, Minneapolis, MN, pp. 3052–3067 (paper ThB08.2)

[5] Doucet A., S.J. Godsill and C. Andrieu, “On sequential Monte Carlo sampling methods for Bayesian filtering”, Statist. Comput., vol. 10, 2000, pp.197-208 .