The SABL Algorithm

John Geweke

August, 2015

Abstract

This is an overview of the SABL algorithm. More detailed material will be avaialble to workshop registrants prior to the workshop

Contents

The SABL algorithm is a procedure for the controlled introduction of new information. It pertains to situations in which information can be represented as the probability distribution of a finite dimensional vector. SABL approximates this distribution by means of many (typically on the order of 10^4 to 10^6) alternative versions of the vector. These versions are called *particles*, reflecting some of SABL's roots in the particle filtering literature. In the SABL algorithm particles undergo a sequence of transformations as information is introduced. With minor exceptions accounting for a negligible fraction of computing time in typical research applications, these transformations amount to identical instructions that operate on each particle in isolation. SABL is therefore a pleasingly parallel algorithm. This property is responsible for dramatic decreases in computing time for many research applications with GPU execution of SABL.

At its highest level the SABL algorithm looks like this:

- Represent initial information
- While information not entirely incorporated
  - Determine information increment and incorporate by weighting particles
  - Remove the weights by resampling
  - Modify the particles to represent the information more efficiently
- End

In the sequential Monte Carlo literature each pass through the loop is known a *cycle*, and we will use ℓ to index cycles. The three steps in each cycle are *phases*. The first step is the *correction phase*, the second the *selection phase*, and the third is the *mutation phase*; for short, C phase, S phase and M phase.

Let $\theta \in \Theta \subseteq \mathbb{R}^d$ denote the vector whose probability distribution represents information. The notation reflects the context of Bayesian inference about a parameter vector. We develop the main ideas in this context and subsequently treat optimization as a variant, in Section 8. Denote the particles by θ_{jn}, the double subscripts indicating J groups of N particles each. Initially θ has probability density $p^0(\theta)$; extension beyond absolutely continuous distributions is easy, and this streamlines the notation. In SABL the particles initially are

$$\theta_{jn}^{(0)} \overset{iid}{\sim} p^{(0)}(\theta) \quad (j = 1, \ldots, J; n = 1, \ldots, N). \tag{1}$$

In Bayesian inference $p^{(0)}(\theta)$ is a proper prior density and in optimization it is the probability density of an instrumental distribution (see Section 8). It must be practical to sample from the initial distribution (1) and to evaluate $p^{(0)}(\theta)$.

Denote the density incorporating all the information by $p^*(\theta)$. SABL requires that it be possible to evaluate a kernel $k(\theta)$ with the properties

$$k(\theta) \geq 0 \;\forall\; \theta \in \Theta, \quad \int_\Theta k(\theta)\, d\theta < \infty, \quad p^*(\theta) \propto k^*(\theta) = p^{(0)}(\theta)\, k(\theta). \tag{2}$$

In Bayesian inference the kernel $k(\theta)$ is the likelihood function,

$$k(\theta) = p(y_{1:T} \mid \theta), \tag{3}$$

where T denotes sample size and $y_{1:T} = \{y_1, \ldots, y_T\}$ denotes the data. In the optimization problem $\max_{\theta \in \Theta} h(\theta)$,

$$k(\theta) = \exp[r \cdot h(\theta)], \tag{4}$$

where $r > 0$ and typically r is large.

Cycle ℓ begins with the kernel $k^{(\ell-1)}$ and ends with the kernel $k^{(\ell)}$. In the first and last cycles,

$$k^{(0)} = 1 \quad \text{and} \quad k^{(L)}(\theta) = k(\theta),$$

respectively. Correspondingly define $k^{*(\ell)}(\theta) = p^{(0)}(\theta)\, k^{(\ell)}(\theta)$, implying

$$k^{*(0)} = p^{(0)}(\theta) \quad \text{and} \quad k^{*(L)}(\theta) = k^{*}(\theta). \tag{5}$$

The particles change in each cycle, and reflecting this let $\theta_{jn}^{(\ell)}$ denote the particles at the end of cycle ℓ. The initial particles $\theta_{jn}^{(0)}$ have the common distribution (1) and are independent. In succeeding cycles the particles $\theta_{jn}^{(\ell)}$ continue to be identically distributed but they are not independent. The theory underlying SABL, discussed further in this section and developed in detail by Durham and Geweke (2015) drawing on sequential Monte Carlo theory, assures that the final particles $\theta_{jn} = \theta_{jn}^{(L)} \xrightarrow{d} p^{*}(\theta)$. This convergence in distribution takes place in N, the number of particles per group. The result is actually stronger: the particles are ergodic in N, meaning that for any function g for which $E[g(\theta)] = \int_{\Theta} g(\theta)\, p^{*}(\theta)\, d\theta$ exists,

$$\lim_{N \to \infty} N^{-1} \sum_{n=1}^{N} g(\theta_{jn}) = E[g(\theta)] \tag{6}$$

with probability 1 in all groups $j = 1, \ldots, J$.

A leading technical challenge in practical sequential Monte Carlo algorithms, which of course work with finite N, is to limit the dependence amongst particles, and in particular to keep dependence from increasing from one cycle to the next to the point that the final distribution of particles is an unreliable representation of any distribution. A further technical challenge is to provide a measure of the accuracy of the approximation implicit in the left side of (6) for finite N that is itself reliable. The SABL algorithm and toolbox do both in a way that makes minimal demands on users. The remainder of this section, and Section **??** that follows, provide the details.

1 C phase

For each cycle ℓ define the weight function

$$w^{(\ell)}(\theta) = k^{(\ell)}(\theta) / k^{(\ell-1)}(\theta).$$

The theory underlying the SABL algorithm requires that there exist an upper bound $\overline{w}^{(\ell)}$, that is,

$$w^{(\ell)}(\theta) < \overline{w}^{(\ell)} < \infty \;\forall\; \theta \in \Theta.$$

The C phase determines $w^{(\ell)}(\theta)$ explicitly and thereby defines

$$k^{(\ell)}(\theta) = w^{(\ell)}(\theta) \cdot k^{(\ell-1)}(\theta). \tag{7}$$

and

$$p^{*(\ell)}(\theta) = k^{*(\ell)}(\theta)\, d\theta / \int_\Theta k^{*(\ell)}(\theta)\, d\theta.$$

Correspondingly, define

$$k^{*(\ell)} = p^{(0)}(\theta)\, k^{(\ell)}(\theta) \tag{8}$$

and note (7) implies $k^{*(\ell)}(\theta) = w^{(\ell)}(\theta) \cdot k^{*(\ell-1)}(\theta)$ as well. The weight functions $w^{(\ell)}(\theta)$ are designed so that there exists $L < \infty$ for which $k^{(L)}(\theta) = k(\theta)$, although the value of L is in general not known at the outset.

One approach is to use the functional form $w^{(\ell)}(\theta) = k(\theta)^{\Delta_\ell}$ and determine a suitable choice of $\Delta_\ell > 0$. Thus at the end of cycle ℓ, $k^{(\ell)}(\theta) = k(\theta)^{r_\ell}$ where $r_\ell = \sum_{s=1}^{\ell} \Delta_s$. This variant of the C phase is known as *power tempering* or simply *tempering*. The term originates in the simulated annealing literature in which $T_\ell = r_\ell^{-1}$ is known as *temperature* and $\{T_\ell\}$ as the *cooling schedule*. Another approach originates in particle filtering and Bayesian inference: $k^{(\ell)}(\theta) = p(y_{1:t_\ell} \mid \theta)$, where $0 < t_1 \ldots < t_L = T$ (sample size). The increments are therefore $w^{(\ell)}(\theta) = p\left(y_{t_{\ell-1}+1:t_\ell} \mid y_{1:t_{\ell-1}}, \theta\right)$. This variant of the C phase is known as *data tempering*.

The C phase can be motivated informally by analogy to importance sampling, a Monte Carlo simulation method with at least a 60-year history, interpreting $k^{*(\ell-1)}(\theta)$ as the kernel of the source density and $k^{*(\ell)}(\theta)$ as the kernel of the target density. (Recall the definition of $k^{*(\ell)}(\theta)$ in (8).) If it were the case that the particles $\theta_{jn}^{(\ell-1)}$ were independent and had common distribution indicated by the kernel density $k^{*(\ell-1)}(\theta)$, then

$$\frac{\sum_{j=1}^{J} \sum_{n=1}^{N} w\left(\theta_{jn}^{(\ell-1)}\right) g\left(\theta_{jn}^{(\ell-1)}\right)}{\sum_{j=1}^{J} \sum_{n=1}^{N} w\left(\theta_{jn}^{(\ell-1)}\right)} \xrightarrow{a.s.} \frac{\int_\Theta k^{*(\ell)}(\theta)\, g(\theta)\, d\theta}{\int_\Theta k^{*(\ell)}(\theta)\, d\theta}$$

$$= \int_\Theta p^{*(\ell)}(\theta)\, g(\theta)\, d\theta = E^{(\ell)}[g(\theta)] \tag{9}$$

so long as $E^{(\ell)}[g(\theta)]$ exists. The convergence is in N, the number of particles per group.

The core of the argument for importance sampling is

$$\int_\Theta p^{*(\ell)}(\theta)\, g(\theta)\, d\theta = \frac{\int_\Theta w^{(\ell)}(\theta)\, k^{*(\ell-1)}(\theta)\, g(\theta)\, d\theta}{\int_\Theta w^{(\ell)}(\theta)\, k^{*(\ell-1)}(\theta)\, d\theta} = \frac{\int_\Theta w^{(\ell)}(\theta)\, (\theta)\, p^{*(\ell-1)} g(\theta)\, d\theta}{\int_\Theta w^{(\ell)}(\theta)\, p^{*(\ell-1)}(\theta)\, d\theta}.$$

This result does not apply strictly, here, because while the particles $\theta_{jn}^{(\ell-1)}$ are identically distributed, they are not independent and $k^{*(\ell-1)}(\theta)$ is at best an approximation of the kernel density of the true common distribution of the particles $\theta_{jn}^{(\ell-1)}$ so long as $N < \infty$ (as it must be in practice). But many of the practical concerns in importance sampling carry over. In particular, success lies in $w(\theta)$ being "well-conditioned" – loosely speaking, variation in $w(\theta_{jn})$ must not be too great. For example, difficulties arise when just a few weights $w(\theta_{jn})$ account for most of the sum. In this case the target density kernel $k^{*(\ell)}(\theta)$ is represented almost entirely by a small number of particles and the approximation of $E^{(\ell)}[g(\theta)]$ implicit in the left side of (9) is poor.

The C phase directly confronts the key question of how much information to introduce in cycle ℓ: too little and L will be larger than it need be; too much, and it becomes difficult for the other phases to convert ill-weighted particles from cycle $\ell-1$ into particles from cycle ℓ sufficiently independent that the representation of the distribution does not deteriorate from one cycle to the next into a state of gross unreliability. A conventional and effective way to monitor the quality of the weight function is by means of *relative effective sample size*

$$RESS^{(\ell)} = \frac{ESS^{(\ell)}}{JN} = \frac{\left[\sum_{j=1}^{J}\sum_{n=1}^{N} w^{(\ell)}\left(\theta_{jn}^{(\ell-1)}\right)\right]^2}{JN\sum_{j=1}^{J}\sum_{n=1}^{N} w^{(\ell)}\left(\theta_{jn}^{(\ell-1)}\right)^2}. \tag{10}$$

The *effective sample size* $ESS^{(\ell)}$ is an adjustment to the sample size (number of particles, JN) that accounts for lack of balance in the weights, and relative effective size is its ratio to sample size. Notice that if all weights are the same then $ESS^{(\ell)} = JN$ and $RESS^{(\ell)} = 1$, whereas if only one weight is positive then $ESS^{(\ell)} = 1$ and $RESS^{(\ell)} = 1/JN$.

In general $RESS^{(\ell)}$ is lower the more information is introduced in the C phase. This is always true for power tempering and as a practical matter is nearly always the case for data tempering. It suggests a strategy of introducing information only up to the point where $RESS^{(\ell)}$ has just dropped or would drop below are target value. The target $RESS^{*} = 0.5$ is usually reasonable, and it is the default value in the SABL toolbox. Practical experience shows that somewhat higher $RESS^{*}$ leads to more cycles but faster execution in the M phase, lower $RESS^{*}$ to fewer cycles but slower M phase execution, and as a result there is not much difference in execution time over the interval $(0.1, 0.9)$ for $RESS^{*}$.

For data tempering this suggests initializing $w^{(\ell)}(\theta) = 1$, followed by iterations $(s = 1, 2, \ldots)$. Iteration s introduces $y_{t_{\ell-1}+s}$, updates

$$w^{(\ell)}\left(\theta_{jn}^{(\ell-1)}\right) = w^{(\ell)}\left(\theta_{jn}^{(\ell-1)}\right)\cdot p\left(y_{t_{\ell-1}+s} \mid y_{t_{\ell-1}+s-1}, \theta_{jn}^{(\ell-1)}\right),$$

and computes the corresponding $RESS^{(\ell)}$. Iterations terminate the first time $RESS^{(\ell)} < RESS^{*}$. This procedure has been well established in the sequential Monte Carlo particle filtering literature for years.

Such strategies have not been employed previously for power tempering. The first instance appears to be Geweke and Frischknecht (2014). Substituting $w^{(\ell)}(\theta) = k(\theta)^{\Delta_\ell}$

in (10),

$$RESS^{(\ell)} = \frac{\left[\sum_{j=1}^{J}\sum_{N=1}^{n} k\left(\theta_{jn}^{(\ell-1)}\right)^{\Delta_\ell}\right]^2}{JN\sum_{j=1}^{J}\sum_{N=1}^{n} k\left(\theta_{jn}^{(\ell-1)}\right)^{2\Delta_\ell}}. \tag{11}$$

Setting $RESS^{(\ell)} = RESS^*$ produces a nonlinear equation in the single variable Δ_ℓ that has a unique and easily computed solution so long as $RESS^* \in (0,1)$. If the solution implies $r^{(\ell)} > 1$ then $\Delta^{(\ell)} = 1 - r^{(\ell-1)}$ instead and the cycle $\ell = L$ is the last one.

2 S phase

The rest of cycle ℓ starts with the weighted particles $\theta_{jn}^{(\ell-1)}$ and produces unweighted particles $\theta_{jn}^{(\ell)}$ that that meet or exceed a mixing condition – a measure of lack of dependence described in the next section. The S phase begins this process, removing weights by means of resampling. The principle behind resampling is to regard the weight function as proportional to a discrete probability function defined over the particles and draw from this distribution with replacement. Hence the name selection phase. SABL performs this operation on each group of particles separately – that is, particles are always selected within groups and never across groups. This independence between the groups $j = 1, \ldots, J$ is essential in (1) proving the convergence of the algorithm, (2) assessing the mixing condition in the M phase, and (3) providing a numerical standard error for the approximation as discussed in Section 6. Resampling produces unweighted particles denoted $\theta_{jn}^{(\ell,0)}$.

The most elementary resampling method is to make N independent and identically distributed draws from the multinomial distribution with argument N and probabilities

$$p_{jn} = w^{(\ell)}\left(\theta_{jn}^{(\ell-1)}\right) / \sum_{i=1}^{N} w^{(\ell)}\left(\theta_{ji}^{(\ell-1)}\right) \quad (n = 1, \ldots, N).$$

This method is known as *multinomial resampling*. An alternative method, known as *residual resampling*, is to compute the same probabilities and collect an initial subsample of size $N^* \leq N$ consisting of $[N \cdot p_{jn}]$ copies of each particle θ_{jn}, where the function $[\cdot]$ is standard notation for what is variously known as the greatest whole integer, greatest integer not greater than, or floor function. Then draw the remaining $N - N^*$ particles by means of multinomial resampling with probabilities probabilities $p_{JN}^* \propto Np_{jn} - [N \cdot p_{jn}]$. Residual resampling results in lower dependence amongst the particles $\theta_{jn}^{(\ell,0)}$ $(n = 1, \ldots, N)$ than does multinomial resampling. For both methods there are central limit theorems that are essential to demonstrating convergence and interpreting numerical standard errors. There are other resampling methods that lead to even less dependence amongst the particles, but for these methods central limit theorems do not apply. These methods are all described in Douc et al. (2005).

The S phase is a simple but key part of the SABL algorithm. Resampling is also a key part of evolutionary (or, genetic) algorithms where it plays much the same role. The particles $\theta_{jn}^{(\ell,0)}$ are for this reason sometimes called the *children* of the *parent* particles $\theta_{jn}^{(\ell-1)}$, and also to emphasize the fact that for each child $\theta_{jn}^{(\ell,0)}$ there is a parent $\theta_{jn'}^{(\ell-1)}$. Parents with larger weights are likely to have more children – it is not hard to work out the exact distribution for any one parent for multinomial resampling and then again for residual resampling. With both, the expected number of children, or fertility, of the parent $\theta_{jn}^{(\ell-1)}$is proportional to $w\left(\theta_{jn}^{(\ell-1)}\right)$, a measure of the parent's "success" in the environment of the information introduced in cycle ℓ.

3 M phase

If the algorithm were to continue in this way, the number of unique children would never increase and in general would decrease from cycle to cycle. Indeed, in the context of Bayesian inference it can be shown under mild regularity conditions that the number of unique children converges almost surely to 1 as the number of observations increases. The same can be demonstrated in the context of optimization for a sufficiently large value of r in (4).

The M phase addresses this problem by creating diversity amongst sibling particles in a way that is faithful to the information kernel $k^{*(\ell)}(\theta)$. It does so using the same principle of invariance that is central to Markov chain Monte Carlo (MCMC) algorithms, drawing particles from a transition density $dQ^{(\ell)}(\theta \mid \theta^*)$ with the property

$$\int_\Theta k^{*(\ell)}(\theta^*)\, dQ^{(\ell)}(\theta \mid \theta^*)\, d\theta^* = k^{*(\ell)}(\theta) \;\; \forall\, \theta \in \Theta. \tag{12}$$

The transition density $dQ^{(\ell)}$ is invariant with respect to the kernel $k^{*(\ell)}(\theta)$, which preserves the original distribution of the children but introduces the prospect that they will be different. Notice that (12) implies that the successive application, or convolution, of a series of invariant transitions defines a transition that is itself invariant. The universe of invariant transition densities is large and manifest in the MCMC literature. Many of these transitions are model-specific, for example Gibbs sampling variants of MCMC. On the other hand a number of families of Metropolis-Hastings transitions apply quite generally and with problem-specific tuning of parameters can be computationally efficient.

The current edition of the SABL toolbox incorporates one of these variants, the Metropolis Gaussian random walk, and the structure of SABL accommodates incorporation of others in the future. The M phase applies the Metropolis random walk repeatedly in steps $s = 1, 2, \ldots$, each step generating a new set of particles $\theta_{jn}^{(\ell,s)}$ from the previous set $\theta_{jn}^{(\ell,s-1)}$. Following the familiar arithmetic, candidate new particles are generated $\theta_{jn}^{*(\ell,s)} \sim N\left(\theta_{jn}^{(\ell,s-1)}, \Sigma^{(\ell,s-1)}\right)$ and accepted with probability

$$\min\left[\frac{k^{*(\ell)}\left(\theta_{jn}^{*(\ell,s)}\right)}{k^{*(\ell)}\left(\theta_{jn}^{(\ell,s-1)}\right)},\ 1\right].$$

In SABL $\Sigma^{(\ell,s)}$ is proportional to the sample variance of $\theta_{jn}^{(\ell,s-1)}$ computed using all the particles. The factor of proportionality increases when the rate of candidate acceptance in the previous step exceeds a specified threshold and is decreased otherwise. This draws on established practice in MCMC and works well in this context. Section **??** provides more detail about this threshold, as well as the initial value and increments of the scaling factor.

In some applications, especially those with a long parameter vector θ, the multivariate normal distribution is a sufficiently poor approximation of the local behavior of $k^{*(\ell)}(\theta)$ that the Metropolis Gaussian random walk can be quite inefficient. A straightforward way to address this contingency is the blocked Metropolis Gaussian random walk variant of the M phase, included in the current edition of SABL. In this variant θ is partitioned into subvectors and the Gaussian random walk Metropolis algorithm is applied to the subvectors in turn. Section **??** provides more detail.

The objective of the M phase is to attain a degree of independence of the particles $\theta_{jn}^{(\ell)}$ at the end of each cycle sufficient to render the final set of particles $\theta_{jn} = \theta_{jn}^{(L)}$ a reliable representation of the distribution implied by the probability density function $p^*(\theta)$. The idea behind M phase termination in SABL is to measure the degree of mixing (lack of dependence) amongst the particles at the send of each Metropolis step s of cycle ℓ, and terminate when this measure meets or exceeds a certain threshold.

In SABL mixing is measured by the average relative numerical efficiency (RNE) of a group of functions chosen specifically for this purpose in each model. The RNE of the SABL approximation of a posterior moment $E[g(\theta)] = \int_\Theta g(\theta)\, p^*(\theta)\, d\theta$ is a measure of its numerical accuracy relative to that achieved by a hypothetical simulation $\theta_{ij} \overset{iid}{\sim} p^*(\theta)$. Section 6 explains how this measure is constructed.

A simple stopping rule for the M phase is to terminate the iterations of the Metropolis random walk when the average RNE of a group of functions first exceeds a stated threshold. In any application there are practical limits to the average RNE that can be achieved through these iterations, and so it is also desirable to impose a limit on their number. Achieving greater independence of particles is especially important in the last cycle, because at the end of the M phase in that cycle the particles constitute the representation of $p^*(\theta)$. There are quite a few options for M phase termination, detailed in Section **??**. The SABL toolbox core default criterion is average RNE 0.4 with 100 maximum iterations in cycles $1, \ldots, L-1$ and average RNE 0.9 with 300 maximum iterations in the final cycle L.

Mixing thoroughly is not the objective of the M phase. In MCMC that is essential in providing a workable representation of the distribution with kernel $k^*(\theta)$. In SABL the C and S phases take on this important task, whereas the function of the M phase is to place a lower bound on the dependence amongst particles. Section **??** introduces some elaborations of this stopping criterion as options in the SABL toolbox.

4 Convergence, the two-pass variant of SABL and accuracy

Durham and Geweke (2015) shows that bounded likelihood

$$\max_{\theta \in \Theta} p\left(y_{1:T} \mid \theta\right) < \infty \tag{13}$$

and existence of the prior moment

$$\int_{\Theta} \left|g\left(\theta\right)\right| p^{(0)}\left(\theta\right) d\theta < \infty \tag{14}$$

respectively are sufficient for the essential condition (6). (Weaker conditions exist but are more difficult to verify: see Durham and Geweke (2015) and references cited there.) In all posterior simulators the assessment of numerical accuracy is based on a central limit theorem, which in this context takes the form

$$N^{1/2}\left(\overline{g}^{(J,N)} - \overline{g}\right) \xrightarrow{d} N\left(0, \sigma_g^2\right) \tag{15}$$

where

$$\overline{g} = \int_{\Theta} g\left(\theta\right) p^{*}\left(\theta\right) d\theta \quad \text{and} \quad \overline{g}^{(J,N)} = N^{-1} \sum_{n=1}^{N} g\left(\theta_{jn}\right).$$

By itself (15) is not enough: it is essential to compute or approximation σ_g^2 as well. Section 6 explains how SABL does this.

5 Convergence and the two-pass variant

The theory developed in the sequential Monte Carlo literature provides a start. It posits a fixed pre-specified sequence of kernels $k^{(1)}, \ldots, k^{(L)}$ (see (7)) and a fixed pre-specified sequence of M phase transition densities $dQ^{(\ell)}$ (see (12)), together with side conditions (implied by conditions (13) and (14)), and proves (15). But in any practical application the kernels $k^{(\ell)}$ and transition densities $dQ^{(\ell)}$ are adaptive, relying on information in the particles $\theta_{jn}^{(\ell-1)}$ or $\theta_{jn}^{(\ell,s-1)}$, rather than fixed. The theory does not apply then because the kernels and transitions depend on the random particles, and the structure of this dependence is so complex as to preclude extension of the existing theory to this case – especially for the transition kernels $dQ^{(\ell)}$. Thus, this literature provides a theory of sequential Bayesian learning but not a theory of *sequentially adaptive* Bayesian learning.

It is universally recognized that some form of adaptation is required, for it is impossible to pre-specify kernels $k^{(\ell)}$ and transition densities $dQ^{(\ell)}$ that provide reliable approximations in tolerable time without knowing a great deal about the posterior distribution – which, of course, is the goal and not the starting point.

Durham and Geweke (2015) deals with this issue by creating the two-pass variant of the algorithm. The first pass is exactly as described in this section, with the addition that the kernels $k^{(\ell)}$ and transitions $dQ^{(\ell)}$ are saved. For the specific variants described in Sections 1 and 3, this amounts to saving the sequence $\{r_\ell\}$ or $\{t_\ell\}$ from the C phase and the doubly-indexed sequence of variance matrices $\Sigma^{(\ell,s-1)}$ from the M phase, but the idea generalizes to other variants of the C and M phases. The second pass re-executes the algorithm (with different seeds for the random number generator) and uses the kernels $k^{(\ell)}$ and transitions $dQ^{(\ell)}$ computed in the first pass, skipping the work required to compute these objects from the particles. The theory developed in the sequential Monte Carlo literature then applies directly to the second pass, because the kernels $k^{(\ell)}$ and transitions $dQ^{(\ell)}$ are in fact fixed in the second pass. The role of the first pass is to provide the knowledge of the posterior distribution required for sensible pre-specification of these objects.

Experience thus far is that substantial differences between the first and second passes do not arise, and can only be made to do so by specifying imprudently small values of N. Thus in practice it suffices to use the two-pass algorithm only occasionally – perhaps at the inception of a research project when the general character of the model(s), data and sample size are known, and then again prior to communicating findings.

6 Numerical accuracy

The sequential Monte Carlo literature provides abstract expressions for σ_g^2 in (15) but no means of evaluating or approximating σ_g^2. SABL provides the approximation using the particle groups. Consider the second pass of the two-pass algorithm where the convergence theory fully applies. In this setting there is no dependence of particles across groups. The M phase and the C phase are perfectly parallel: exactly the same operations applied to all the particles with no communication between particles. Resampling in the S phase, which introduces dependence amongst particles, takes place entirely within groups so as not to compromise independence across groups. Therefore the approximations $\overline{g}_{jN} = N^{-1}\sum_{n=1}^{N} g(\theta_{jn})$ of $\overline{g} = E[g(\theta)]$ are independent across the groups $j = 1, \ldots, J$. A central limit theorem (15) applies within each group. Computing the cross-group mean $\overline{g}_{J,N} = J^{-1}\sum_{j=1}^{J}\overline{g}_{jN}$, a conventional estimate of σ_g^2 in (15) is

$$\widehat{\sigma}_g^2 = N \cdot (J-1)^{-1} \sum_{j=1}^{J} \left(\overline{g}_{jN} - \overline{g}_{J,N}\right)^2 \tag{16}$$

and

$$(J-1)\,\widehat{\sigma}_g^2/\sigma_g^2 \xrightarrow{d} \chi^2(J-1), \tag{17}$$

the convergence in (17) being in particles per group N. In the limit $N \to \infty$, $\overline{g}_{J,N}$ and $\widehat{\sigma}_g^2$ are independent.

The corresponding numerical variance estimate for $\overline{g}_{J,N}$ is

$$\widehat{\sigma}_{g,JN}^2 = (JN)^{-1}\,\widehat{\sigma}_g^2 \tag{18}$$

and the numerical standard error is $\widehat{\sigma}_{g,JN} = \left(\widehat{\sigma}_{g,JN}^2\right)^{1/2}$. This should not be confused with the approximation of the posterior variance,

$$\widehat{\text{var}}\,(g) = (JN)^{-1} \sum_{j=1}^{J} \sum_{n=1}^{N} \left[g\left(\theta_{jn}\right) - \overline{g}_{J,N}\right]^2 .$$

The *numerical standard error* corresponding to (18) is $\widehat{\sigma}_{g,JN} = \left[\widehat{\sigma}_{g,JN}^2\right]^{1/2}$. This is the measure of accuracy used in SABL. From (17) the formal interpretation of numerical standard error is

$$\frac{\overline{g}_{J,N} - \overline{g}}{\widehat{\sigma}_{g,JN}} \xrightarrow{d} t\,(J-1)\,.$$

If particles within groups are independent then $\widehat{\sigma}_g^2 \approxeq \widehat{\text{var}}\,(g)$, whereas if they are not then usually $\widehat{\sigma}_g^2 > \widehat{\text{var}}\,(g)$, although $\widehat{\sigma}_g^2 < \widehat{\text{var}}\,(g)$ may occur and is more likely with smaller numbers of particle groups J. The *relative numerical efficiency* of the approximation $\overline{g}_{J,N}$ is

$$RNE_g = \widehat{\text{var}}\,(g)\,/\widehat{\sigma}_g^2. \tag{19}$$

A useful interpretation of (19) is a hypothetical simulator with $\theta_{jn} \overset{iid}{\sim} p^*\,(\theta)$ would achieve the same accuracy with $RNE_g \cdot JN$ particles.

This argument does not apply directly in the first pass because of the adaptation. In particular, recall that RNE is used in the M phase to assess mixing and determine the end of the sequence of iterations of the Metropolis random walk. This is an example of the complex feedback between particles and adaptation in the algorithm that has frustrated central limit theorems. This shortfall in theory is likely to persist. The two-pass procedure overcomes the problem and, moreover, provides the foundation for future variants of the algorithm without the overhead of establishing convergence for each variant.

7 Marginal likelihood

The SABL algorithm is particularly well suited to providing a numerical approximation of the marginal likelihood

$$ML = \int_\Theta p^{(0)}\,(\theta)\,p\,(y_{1:T} \mid \theta)\,d\theta = \int_\Theta k^*\,(\theta)\,d\theta. \tag{20}$$

The marginal likelihood, also called the marginal data density, is central in the theory and practice of Bayesian model comparison, as well as in Bayesian model averaging for combining models and decision-making. It has posed a particularly difficult technical problem that has seen checkered resolution in the posterior simulation literature as well as in practice: depending on the combination of posterior simulation method and model, approximation of ML can be easy to impossible, and reliably assessing the accuracy of the approximation poses further issues that are again specific to the situation.

The SABL algorithm produces approximations of ML – more precisely $\log ML$ as is standard – as a by-product of the C phase. Here we will present the ideas behind the method, without going into full detail which requires considerable additional notation. Details are in Section 4 of Durham and Geweke (2015) and are reflected in the SABL toolbox code. From (5) and (7),

$$\begin{aligned}\int_{\Theta} k^*(\theta)\,d\theta &= \frac{\int_{\Theta} k^{*(L)}(\theta)\,d\theta}{\int_{\Theta} k^{\,*(0)}(\theta)\,d\theta} = \prod_{\ell=1}^{L} \frac{\int_{\Theta} k^{*(\ell)}(\theta)\,d\theta}{\int_{\Theta} k^{*(\ell-1)}(\theta)\,d\theta} \\ &= \prod_{\ell=1}^{L} \frac{\int_{\Theta} w^{(\ell)}(\theta)\,k^{*(\ell-1)}(\theta)\,d\theta}{\int_{\Theta} k^{*(\ell-1)}(\theta)\,d\theta} = \prod_{\ell=1}^{L} \int_{\Theta} w^{(\ell)}(\theta)\,p^{(\ell-1)}(\theta)\,d\theta. \qquad (21)\end{aligned}$$

In the C phase of cycle ℓ, as $N \to \infty$,

$$\overline{w}_{\ell,J,N} = (JN)^{-1} \sum_{j=1}^{J} \sum_{n=1}^{N} w^{(\ell)}\left(\theta_{jn}^{(\ell-1)}\right) \xrightarrow{a.s.} \int_{\Theta} w^{(\ell)}(\theta)\,p^{(\ell-1)}(\theta)\,d\theta,$$

Hence from (21),

$$\prod_{\ell=1}^{L} \overline{w}_{\ell,J,N} \xrightarrow{a.s.} \int_{\Theta} k^*(\theta)\,d\theta,$$

where the convergence is again in the number of particles per group N. This is the marginal likelihood (20) in a Bayesian inference context. Durham and Geweke (2015) discusses the approximation of $\log(ML)$ and computing the numerical standard error for that approximation.

8 Optimization

Return now to the optimization problem, determining $\theta^* = \arg\max_{\theta \in \Theta} h(\theta)$. As discussed in Section **??**, SABL approaches this problem using kernels of the form (4) in a manner analogous to the likelihood function $p(y_{1:T} \mid \theta)$ in Bayesian inference. There continues to be an initial density $p^{(0)}(\theta)$, and the corresponding distribution is sometimes call the instrumental distribution in this context. This might or might not be intended or interpreted as the expression of prior beliefs about the solution of the optimization problem. The density $p^{(0)}(\theta)$ is a technical device providing an initial condition for the algorithm and requires only that $p^{(0)}(\theta) > 0 \;\forall\; \theta \in \Theta$.

If $h(\theta)$ is bounded above on Θ (the analogue of (13)) then SABL produces particles θ_{ij} whose distribution has kernel density $p^{(0)}(\theta)\exp[r \cdot h(\theta)]$. If h has a unique global mode θ^* then, under weak side conditions stated in Geweke and Frischknecht (2014), $\theta \xrightarrow{p} \theta^*$ as $r \to \infty$. In practice one does not know values of r required for satisfactory results. Recall that in the power tempering variant of the C phase the sequence $\{r_\ell\}$ is derived from relative effective sample size (11), a function of the particles $\theta_{ij}^{(\ell-1)}$. Replace the termination criterion $r^{(L)} = 1$ used for Bayesian inference with one that defines a suitably close approximation of θ^*.

An example of such a termination criterion is an upper bound for the range of values of $h(\theta_{ij})$ over all JN particles. (As discussed in Section **??**, there are other convergence criteria as well, and it is easy for users to provide a customized convergence criterion.) What is a reasonable threshold size is problem-specific but usually clear. If h is denominated in dollars then a range of one dollar is likely more than adequate. For maximum likelihood the kernels are $p(y_{1:T} \mid \theta)$, equivalently $h(\theta) = \log p(y_{1:T} \mid \theta)$, and a range of $h(\theta_{ij})$ somewhere between 10^{-2} and 10^{-8} is likely adequate.

It is typical to see a steady increase in power with each successive cycle. In fact for twice continuously differentiable objective functions $h(\theta)$ it can be shown that the rate of change $(r_\ell - r_{\ell-1})/r_{\ell-1}$ comes to depend only on the number of elements in θ. If the C phase power tempering criterion is $RESS^* = 0.5$ then for θ with 3 elements it is 1.153, 10 elements 0.56, 20 elements 0.349, and 50 elements 0.198. Values are lower for higher $RESS^*$ and vice versa. Observed rates are often very close to the theoretical values in most cycles. Eventually, however, this breaks down because the particles θ_{ij} are very close and differences amongst the $h(\theta_{ij})$, which are critical in evaluating $RESS$, become dominated by rounding error arising from the finite number of bits in floating point arithmetic. At this point the implementation of the algorithm is no longer faithful to its analytical properties and to continue is to produce noise. As with any other convergence algorithm it is productive to select explicit convergence explicitly and thoughtfully.